DYNAMICS AND CORRELATIONS IN SPARSE SIGNAL ACQUISITION

A Dissertation
Presented to
The Academic Faculty

By

Adam Shabti Charles

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
May 2015

Copyright © 2015 by Adam Shabti Charles

DYNAMICS AND CORRELATIONS IN SPARSE SIGNAL ACQUISITION

Approved by:

Dr. Christopher J. Rozell, Advisor
Associate Professor, School of ECE
Georgia Institute of Technology

Dr. David Anderson
Professor, School of ECE
Georgia Institute of Technology

Dr. Justin K. Romberg
Associate Professor, School of ECE
Georgia Institute of Technology

Dr. Byron Boots
Assistant Professor, School of Interactive Computing
Georgia Institute of Technology

Dr. Mark A. Davenport
Assistant Professor, School of ECE
Georgia Institute of Technology

Date Approved: March, 13 2015

This work is dedicated to all the educators who have taught me and countless others to love learning and have lead me to this point.

“I’ve measured it from side to side: ’Tis three feet long, and two feet wide.”

– William Wordsworth, The Thorn, 1798

ACKNOWLEDGMENT

This work has been the culmination of many years of work with the help of a multitude of collaborators, friends and family. I would like to extend a heart-felt thank you to all those who have allowed me to climb the academic mountain in order to be able to construct this thesis and the works therein.

First and foremost I would like to thank my advisor and mentor Dr. Christopher Rozell. I showed up to Georgia Tech very much in the rough. Dr. Rozell taught me the intricacies of navigating the academic world, from designing and answering important research questions to learning how to communicate my findings to the rest of the academic world. I know that my work and my path through life would be quite different had I not joined Dr. Rozell's lab. I would also like to thank my committee members: Dr. Justin Romberg, Dr. Mark Davenport, Dr. David Anderson and Dr. Byron Boots for all their input and patience.

As with most researchers, I have been influenced by the wonderful researchers around me, and I would like to thank all my collaborators: Dr. Justin Romberg, Dr. Mark Davenport, Dr. Bruno Olshausen, Dr. Nicholas Tufillaro, Dr. Jennifer Hasler, Dr. Ali Ahmed, Dr. Salman Asif, Dr. Han Lun Yap, Dr. Aurele Balavoine, Dr. Christopher Turnes, Dr. Pierre Garrigues, Dr. Abigail Kressner, Steven Connover, Dr. Samuel Shapero, Mengchen Zhu, Marissa Norko and Dong Yin. Every single one of these collaborators have helped me learn new concepts and methods, allowing me to expand into a better, more well rounded researcher. I would like to make an special mention of the wonderful lab group I was privileged to work with: Dr. Han Lun Yap, Dr. Samuel Shapero, Abbigail Kressner, Mengchen Zhu, Nicholas Bertrand, Marissa Norko, Pavel Dunn, Allison DelGiorno, Dhruv Vishwakarma, and Dong Yin. I have worked, joked and discussed many important and unimportant topics with these amazing peers, and I feel that I have been all the better for it.

As interest in research often starts before graduate school, I would like to thank a number of my college professors for encouraging me to pursue a carreer in research in electrical

engineering: Dr. Fred Fontaine, Dr. Toby Cumberbatch, Dr. Hamid Ahmed and Dr. Robert Uglesich. Without the educational influences of these professors, I would not have progressed to far in the electrical engineering discipline.

In addition to all the support I have received from my academic peers, I have also received tremendous support from my friends and family. My friends, Dr. Jeff Bingham, Dr. Sean Kelly, Dr. Taylor Shapiro, Dr. Brandon Sforzo, Gina Magnotti, Andrew Massimino, and Michael Moore, among many others, have constantly encouraged me during my years in Atlanta. My family has always supported my education, and in particular, I would like to thank my parents Dr. Richard and Shulamit Charles and sister Aya Grant for always investing in my education and grand uncle Louis Charles for his interest and encouragement in my academic pursuits. I would also like to thank Dr. Jack Lowenthal, who has been like to family to me, always showing interest in my studies and constantly lending me physics and mathematics books.

Last, but by no means least, I would like to thank my lovely wife Diane for her support and patience with me as I fretted for hours over papers, presentations, proofs, reading, and all the other multitudes of tasks asked of graduate students.

TABLE OF CONTENTS

ACKNOWLEDGMENT v

LIST OF FIGURES x

SUMMARY xiv

I INTRODUCTION 1

II BACKGROUND 9
2.1 Sparse Signals and Compressive Sensing 9
2.2 Re-weighted ℓ_1 13
2.3 Dictionary Learning 16

III SHORT-TERM MEMORY IN ECHO-STATE NETWORKS 20
3.1 STM Capacity using the RIP 23
3.1.1 Network Dynamics as Compressed Sensing 23
3.1.2 Single Finite-Length Input 23
3.1.3 Sparse Multiple Finite-Length Inputs 29
3.1.4 Low-Rank Multiple Finite-Length Inputs 32
3.1.5 STM Capacity of Infinite-Length Inputs 36
3.2 Other Network Constructions 39
3.2.1 Alternate Orthogonal Constructions 39
3.2.2 Suboptimal Network Constructions 42
3.3 Discussion 44

IV TRACKING OF TIME-VARYING SIGNALS 45
4.1 Basis Pursuit De-Noising with Dynamic Filtering 49
4.1.1 Optimization Framework for State Estimation 49
4.1.2 Sparsity in the Dynamics 50
4.1.3 Simulations 53
4.2 Guarantees on Basis Pursuit De-Noising with Dynamic Filtering 56
4.2.1 General Convergence Guarantees 57
4.2.2 ISTA based convergence 59
4.3 Re-Weighted ℓ_1 Dynamic Filtering 62
4.4 Dynamic Filtering Simulations 66
4.4.1 Stylized tracking scenario 67
4.4.2 CS recovery of natural video sequences 70
4.5 Learning Dynamics Functions 74
4.5.1 Learning a Bilinear BPDN-DF Model 75
4.5.2 Learning a Bilinear RWL1-DF Model 78

V SPATIALLY CORRELATED INFERENCE IN HYPERSPECTRAL IMAGERY . . . 92
5.1 Hyperspectral Imagery . . . 92
5.2 Background and Related Work . . . 97
5.2.1 Methods . . . 97
5.2.2 Hyperspectral dataset and learned dictionaries . . . 100
5.2.3 Related work . . . 101
5.3 Analyzing the Learned Dictionary . . . 102
5.3.1 Learned Dictionary Functions . . . 102
5.3.2 Reconstructing HSI-resolution from MSI-resolution data . . . 108
5.3.3 Supervised classification . . . 117
5.4 Re-weighted ℓ_1 Methods for Spectral Super-resolution . . . 122
5.4.1 Reweighted ℓ_1 Spatial Filtering (RWL1-SF) . . . 123
5.4.2 Performance Comparisons . . . 125
5.5 Applications to Oceanic Imagery . . . 127
5.6 Discussion . . . 129

VI NETWORK ARCHITECTURES FOR ANALOG OPTIMIZATION . . . 135
6.0.1 Sparse Coding . . . 137
6.0.2 Dynamical systems for ℓ_1 minimization . . . 138
6.1 CS Recovery via the LCA . . . 140
6.1.1 BPDN optimization through the LCA . . . 140
6.1.2 LCA solution quality . . . 142
6.1.3 LCA convergence time . . . 145
6.1.4 MRI Reconstruction . . . 147
6.2 Alternate inference problems in the LCA architecture . . . 149
6.2.1 Approximate ℓ_p norms ($0 \leq p \leq 2$) . . . 151
6.2.2 Modified ℓ_p norms . . . 153
6.2.3 Block ℓ_1 . . . 158
6.2.4 Re-weighted ℓ_1 and ℓ_2 . . . 159
6.3 Discussion . . . 161

VII CONCLUSIONS AND FUTURE WORK . . . 164

VIII APPENDICES . . . 169
8.1 Bayesian Approach to Kalman Filtering . . . 169
8.2 General Temporal Convergence for BPDN-DF . . . 173
8.3 ISTA-based Temporal convergence for BPDN-DF . . . 176
8.4 RIP for Single Input with Optimal Feed-Forward Vectors . . . 180
8.5 RIP for Single Input with Gaussian Feed-Forward Vectors . . . 185
8.6 RIP with Multiple Sparse Inputs . . . 194
8.7 RIP for Multiple Low-Rank Inputs . . . 199
8.7.1 Matrix Bernstein Inequality and Olicz Norm . . . 217
8.8 Derivation of recovery bound for infinite length inputs . . . 219

REFERENCES 222

LIST OF FIGURES

Figure 1 Block diagram of sensing systems. 3

Figure 2 Depiction of the restricted isometry property. 11

Figure 3 Echo-state network dynamic evolution. 21

Figure 4 Single input with a sparse structure. 24

Figure 5 Example short-term memory recovery. 28

Figure 6 Phase-transition diagrams for short-term memory recovery. 29

Figure 7 Multiple inputs with a sparse structure. 30

Figure 8 Logarithmic dependence on the number of inputs for short-term memory. 33

Figure 9 Multiple inputs with a low-rank structure. 34

Figure 10 Optimal recovery length for infinite length inputs into echo-state networks. 39

Figure 11 Different connectivity patterns for orthogonal networks. 41

Figure 12 Norm-based depiction of different dynamic filtering algorithms. 46

Figure 13 Temporal convergence of BPDN-DF for Gaussian innovations and sparse states. 54

Figure 14 Steady-state error behavior of BPDN-DF for sparse innovations and sparse states. 55

Figure 15 Steady-state behavior of BPDN-DF for sparse innovations and sparse states. 56

Figure 16 Temporal convergence of BPDN-DF for sparse innovations and Gaussian states. 57

Figure 17 Comparison of empirical and theoretical behavior of BPDN-DF. 59

Figure 18 Graphical model for RWL1-DF. 62

Figure 19 Algorithmic convergance for RWL1 and RWL1-DF. 69

Figure 20 Tracking results for BPDN-DF and RWL1-DF for synthetic tracking simulations. 86

Figure 21 Compressive sensing recovery of the full Foreman video sequence using BPDN-DF and RWL1-DF. 87

Figure 22 Comparison of RWL1-DF with state-of-the-art tracking algorithms for compressive sensing recovery of video. 88

Figure 23 Histogram comparison of dynamic filtering algorithms for compressive recovery of video. 88

Figure 24 Percent improvement of RWL1-DF over other dynamic filtering algorithms for compressive recovery of video. 89

Figure 25 The dynamics learning algorithm can learn an identity basis and a simple permutation dynamics function. 90

Figure 26 The dynamics learning algorithm can learn an identity basis and a set of permutation dynamics functions. 90

Figure 27 Improved recovery of video patches using learned dictionaries for dynamics. 91

Figure 28 Manifold depiction of typical endmember analysis and sparse-coding based analysis. 96

Figure 29 Example spectra for labeled materials in the Smith Island dataset and similar learned dictionary elements. 103

Figure 30 PCA vs. sparse coding for HSI analysis. 104

Figure 31 Progression of sparse coding coefficients from a row of contiguous pixels in the Smith Island dataset. 105

Figure 32 Scatter plot of the nonlinar water pixel manifold and corresponding learned water spectra. 107

Figure 33 Reconstructing spectra with HSI resolution from measurements with MSI-level resolution. 109

Figure 34 Reconstructing HSI data from simulated coarse HSI measurements using training and testing data collected on the same date. 112

Figure 35 Reconstructing HSI data from simulated coarse HSI measurements using training and testing data collected on different dates (in different seasons). 113

Figure 36 Reconstructing HSI data from simulated MSI measurements using training and testing data collected on the same date. 114

Figure 37 Reconstructing HSI data from simulated MSI measurements using training and testing data collected on different dates (in different seasons). . . 116

Figure 38 Relationship of the normalized sparsity $\|\boldsymbol{a}\|_1/\|\boldsymbol{a}\|_2$ of the coefficients to the reconstruction errors when inferring HSI-resolution data from simulated MSI measurements. 117

Figure 39 Vector Quantization classification in the Smith Island dataset. 119

Figure 40 Classification on 22 material classes in the Smith Island dataset. 120

Figure 41 Depiction of the kernel $\boldsymbol{K}$ that determines the influence of neighboring pixels on inference at a given location. 125

Figure 42 Performance improvement and error distribution of RWL1-SF applied to the Smith Island dataset. 128

Figure 43 Example super-resilution improvement using RWL1-SF. 129

Figure 44 Improved recovery of HSI using RWL1-SF via PCA analysis of recovered data. 130

Figure 45 Blurring matrix which transforms HICO spectra into VIIRS spectra. . . . 131

Figure 46 Examples of spectral super-resolution of a VIIRS image taken around the Acqua Alta Oceanographic Tower (AAOT) near Venice, Italy on February 11, 2012. 132

Figure 47 HICO level super-resolution from VIIRS samples. 133

Figure 48 Spatial heat map of super-resolution errors in the AAOT dataset. 134

Figure 49 Log barrier relaxations of BPDN. 141

Figure 50 Phase transition diagrams comparing LCA solutions with digital solvers for BPDN. 143

Figure 51 Temporal convergence of the simulated LCA compared to GPSR. 144

Figure 52 Convergence behavior for the simulated LCA for a number of different problem sizes. 146

Figure 53 Reconstruction of 256x192 pixel MR images from simulated CS acquisition using LCA and YALL1. 148

Figure 54 Various cost function and corresponding LCA thresholding functions. . . 151

Figure 55 Approximate ℓ_p cost functions and their corresponding thresholding functions. 152

Figure 56 Block-sparse LCA threshold function. 159

Figure 57 Re-weighted ℓ_1 optimization via the LCA. 162

Figure 58 Norm-based depiction of the Kalman figure procedure. 170

SUMMARY

One of the most important parts of engineered and biological systems is the ability to acquire and interpret information from the surrounding world accurately and in time-scales relevant to the tasks critical to system performance. This classical concept of efficient signal acquisition has been a cornerstone of signal processing research, spawning traditional sampling theorems (e.g. Shannon-Nyquist sampling), efficient filter designs (e.g. the Parks-McClellan algorithm), novel VLSI chipsets for embedded systems, and optimal tracking algorithms (e.g. Kalman filtering). Traditional techniques have made minimal assumptions on the actual signals that were being measured and interpreted, essentially only assuming a limited bandwidth. While these assumptions have provided the foundational works in signal processing, recently the ability to collect and analyze large datasets have allowed researchers to see that many important signal classes have much more regularity than having finite bandwidth.

One of the major advances of modern signal processing is to greatly improve on classical signal processing results by leveraging more specific signal statistics. By assuming even very broad classes of signals, signal acquisition and recovery can be greatly improved in regimes where classical techniques are extremely pessimistic. One of the most successful signal assumptions that has gained popularity in recet hears is notion of sparsity. Under the sparsity assumption, the signal is assumed to be composed of a small number of atomic signals from a potentially large dictionary. This limit in the underlying degrees of freedom (the number of atoms used) as opposed to the ambient dimension of the signal has allowed for improved signal acquisition, in particular when the number of measurements is severely limited.

While techniques for leveraging sparsity have been explored extensively in many contexts, typically works in this regime concentrate on exploring *static* measurement systems

which result in *static* measurements of *static* signals. Many systems, however, have non-trivial dynamic components, either in the measurement system's operation or in the nature of the signal being observed. Due to the promising prior work leveraging sparsity for signal acquisition and the large number of dynamical systems and signals in many important applications, it is critical to understand whether sparsity assumptions are compatible with dynamical systems. Therefore, this work seeks to understand how dynamics and sparsity can be used jointly in various aspects of signal measurement and inference.

Specifically, this work looks at three different ways that dynamical systems and sparsity assumptions can interact. In terms of measurement systems, we analyze a dynamical neural network that accumulates signal information over time. We prove a series of bounds on the length of the input signal that drives the network that can be recovered from the values at the network nodes [1–9]. We also analyze sparse signals that are generated via a dynamical system (i.e. a series of correlated, temporally ordered, sparse signals). For this class of signals, we present a series of inference algorithms that leverage both dynamics and sparsity information, improving the potential for signal recovery in a host of applications [10–19]. As an extension of dynamical filtering, we show how these dynamic filtering ideas can be expanded to the broader class of spatially correlated signals. Specifically, explore how sparsity and spatial correlations can improve inference of material distributions and spectral super-resolution in hyperspectral imagery [20–25]. Finally, we analyze dynamical systems that perform optimization routines for sparsity-based inference. We analyze a networked system driven by a continuous-time differential equation and show that such a system is capable of recovering a large variety of different sparse signal classes [26–30].

CHAPTER I

INTRODUCTION

Signal acquisition and estimation is a vital first step in many engineering applications. For instance, in magnetic resonance imaging (MRI) data acquisition, the MR images are sampled in k-space (spatial frequency space) and need to be transformed into an image prior before physicians can make a diagnosis. Likewise, in remote sensing applications we must first measure terrestrial spectral signatures and estimate their material compositions before useful tasks such as target identification or anomaly detection can be be accomplished. Measuring and interpreting signals is a complex process with many aspects that must be considered simultaneously. In this work we consider the acquisition of a signal and extraction of the useful information contained therein in three parts: observation, estimation, and implementation. While these tasks are by no means independent of one another, we use this partitioning to express signal acquisition via the following three questions:

- How can we measure our signal and how can we quantify the measurements' quality?
- How can we estimate our signal from our measurements?
- How can we compute our estimate quickly and efficiently?

Figure 1 shows the block diagram depicting how the different aspects in this measurement process partitioning interact with each other. In the first question we discuss how our signal of interest $\boldsymbol{x} \in \mathbb{R}^N$ is observed, and what quality these measurements have. In particular, $\boldsymbol{x}$ is rarely observed perfectly, as even the most accurate measuring devices still result in noisy measurements. These noisy observations, $\boldsymbol{y} \in \mathbb{R}^M$, can often be modeled as the inner products of the signal with M measurement vectors,

$$\boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{\epsilon}, \tag{1}$$

where $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ is the measurement matrix whose rows are the measurement vectors, and $\boldsymbol{\epsilon} \in \mathbb{R}^{M}$ is the observation noise. In this first line of inquiry, we try to understand what our measurement vectors $\mathbf{\Phi}$ imply for our recovery process by quantifying how $\mathbf{\Phi}$ preserves the information of $\boldsymbol{x}$ as well as how robust this process is to the noise term $\boldsymbol{\epsilon}$. The quality of these measurements is typically quantified by properties of the matrix $\mathbf{\Phi}$. For instance, if the measurement vectors are too similar, then the signal space is not well sampled, and signals not in the space spanned by the rows of $\mathbf{\Phi}$ will not be observable.

In the second question we look at estimation or inference methods that can recover our signal, or at least the relevant signal statistics, from our measurements. In particular, we must devise an estimator that translates our measurements into a signal estimate, $\widehat{\boldsymbol{x}}$, that is fit for later use. For example, k-space measurements obtained in MRI devices, no matter how accurate, are useless to a physician if not translated from the frequency domain into a spatial image. Most estimation algorithms accomplish this task by designing and optimizing a cost function

$$\widehat{\boldsymbol{x}} = \arg\min_{x} J(\boldsymbol{x}; \boldsymbol{y}, \mathbf{\Phi}), \tag{2}$$

where the cost function $J : \mathbb{R}^{N} \to \mathbb{R}$ is a function of our measurement model parameters and measurements. This cost function is often represented as a sum of terms that represent a combination of our prior knowledge of the signal and our confidence in our measurements. This step is extremely important in situations where the quality of the measurements, with respect to the noise model, is sub-par. *A priori* knowledge of the signal statistics can make up for high levels of deficiencies in the measurements, including highly incomplete measurements as well as high levels of noise.

While often simple to design, optimizing a particular cost function can be very computationally burdensome. The third question considers the actual optimization implementation and addresses the efficiency of various methods to solve Equation (2) for different classes of cost functions. In particular, the third question considers the computational cost,

the generality in terms of the types of cost functions that are solvable, and theoretical guarantees on convergence times and solution accuracy for these systems and algorithms.

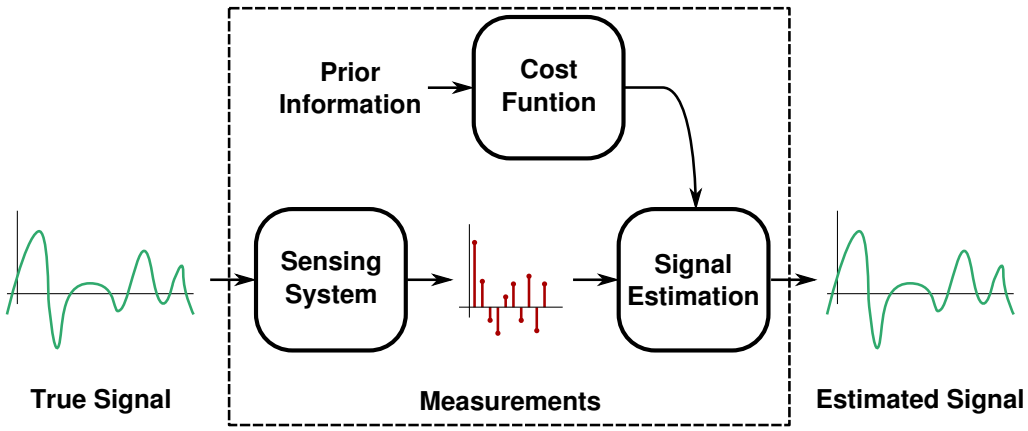

Figure 1: Block diagram of a general sensing paradigm. The incoming signal is observed by the sensing block to produce a vector of measurements. The measurements are then used in an estimation algorithm designed using prior information on the signal to recover an estimate of the original signal.

Recent advances in signal processing have dramatically improved the efficiency of the measurement process, both in terms of the number of measurements needed to recover high dimensional signals as well as the efficiency of the recovery algorithms, when the signal of interest has known statistics. One prominent example used in the fields of *compressive sensing* [31] and *sparse coding* [32] assumes that signals have many fewer degrees of freedom as compared to the ambient dimension. In this case, the signal is said to be *compressible* in that any $\boldsymbol{x}$ is composed by a generative model

$$\boldsymbol{x} = \sum_{i=1}^{N_2} \boldsymbol{\psi}_i \boldsymbol{a}[i] = \boldsymbol{\Psi}\boldsymbol{a},$$

where $\boldsymbol{\psi}_i$ for $i \in [1, N_2]$ are a set of dictionary elements that combine linearly to form the signal $\boldsymbol{x}$, and the vector of linear coefficients $\boldsymbol{a} \in \mathbb{R}^{N_2}$ is compressible in that the energy in $\boldsymbol{a}$ is concentrated on at most $S \ll N$ elements. $\boldsymbol{x}$ is called S-sparse if the energy outside of those S elements of $\boldsymbol{a}$ is zero. This model, while seemingly abstract, has proven relevant in a host of applications, from remote sensing to neuroscience [20, 32]. In cases where the

generative model is suspected to hold, but the dictionary $\mathbf{\Psi}$ is not known *a priori* unsupervised dictionary learning procedures can use example signals from a given class of signals to learn the underlying dictionary [32, 33]. Given these statistics, many cost functions have been designed to recover the signal by recovering the sparse coefficients. One of the simplest and most effective methods for sparse signal estimation is basis-pursuit de-noising (BPDN), which solves the optimization program

$$\widehat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}} \|\boldsymbol{y} - \mathbf{\Phi}\mathbf{\Psi}\boldsymbol{a}\|_2^2 + \gamma\|\boldsymbol{a}\|_1, \tag{3}$$

where $\|\boldsymbol{a}\|_1 = \sum_i |\boldsymbol{a}[i]|$ is the ℓ_1 norm, γ is a parameter that trades off between the ℓ_2 measurement fidelity term and the ℓ_1 sparsity-inducing term, and the estimate is recovered by $\widehat{x} = \mathbf{\Psi}\widehat{\boldsymbol{a}}$. In addition to improved recovery performance from few measurements ($M < N$), a number of highly efficient algorithms, both digital and analog, have been created to quickly recover the signal estimate [34, 35].

While sparsity-based models have greatly increased the ability and efficiency of sensing systems, these systems do not account for many signals and systems that have significant additional structure. Specifically a number of systems depend on dynamically evolving quantities that may be independent of any sparsity assumptions. For example, there is an increased interest in recursive neural networks as information accumulators for decision-based tasks that operate on the information stored in the network nodes. This dynamic network model is essentially a dynamic measurement system that integrates signal information over time into a set of network node values that can be considered measurements of the signal. As opposed to networks that accumulate information to generate measurements, another class of temporally evolving networks actually perform computational tasks, taking in measurements and converging over time to an estimate of the signal which produced those measurements. These implementations, which use temporally evolving circuits for signal estimation, have become a popular viewpoint both for understanding the computational tasks performed in biological neural networks, and laying the foundation for fast analog solvers which could be implemented in real-time embedded systems [27, 35–38].

Correct utilization of such systems, both for measuring and inferring signals, hinges on understanding the capabilities and limitations of such networks. For example the correlations induced between effective measurement vectors in dynamic measurement systems might limit the signal classes that can be effectively observed. Likewise, dynamic inference algorithms tied to certain realistic processes may be less flexible than digital algorithms implemented on general computing hardware. Therefore, to make use of dynamic systems, we need to understand how dynamic systems affect or are affected by signal structure.

As another example, many signal classes have non-trivial dynamic or spatial correlations stemming from set physical process. For instance, the dynamics could dictate a signal's evolution through time, or physical constraints on material distributions in terrestrial imaging can imbue a signal with spatial regularity. In this proposal we will addresses specific instances where accounting for dynamics or correlations could further improve sparse signal acquisition. Specifically we address a modified version of the original three acquisition questions:

- How can a dynamical system measure sparse signals and can we assess the quality of those measurements?
- How can we efficiently and robustly recover dynamically or spatially correlated sparse signal from our measurements?
- How can a dynamic system solve sparsity-inducing optimization programs quickly and efficiently?

This work seeks answers to these three questions in order to expand on the applicability of structured signal assumptions to problems with additional dynamic structure. In particular we divide this work up into four main research directions. Three of these directions are directly related to these three questions, and the fourth demonstrates that the tools derived for dynamic systems can be used for a broader class of spatially correlated signals. We will

demonstrate in this work that many dynamical systems can be used in conjunction with modern sparsity-based signal processing techniques.

In Chapter 3 we address the first of these questions by considering the dynamic echo-state network (ESN), which can measure streaming signals over time. ESNs, networks constructed of random recurrent connections between nodes, are a vital tool in the neural network literature as the computational abilities of such networks are useful in prediction and classification tasks. As opposed to deep learning networks, ESNs make use of recursive connections to obtain far richer dynamics. Additionally, as opposed to trained recursive networks, ESNs make use of random connections, bypassing the computational difficulties in training recursive connections [39, 40]. ESNs have been used for tasks such as speech recognition [41–43], motion detection [44, 45], event detection [46, 47], and noise modeling [48]. This computational utility of the node values imply that the network is integrating the input sequences over time, forming a set of node values that contain important information about the inputs. The work in Chapter 3 explores ESNs as a measurement system, determining how the node values in the network are influenced over time by the input signal. Specifically we quantify the measurement quality by deriving non-asymptotic bounds on the length of the input sequences that the network nodes (i.e. the measurements) can be used to recover (a quantity known as the short-term memory (STM) of the network). The STM of the network gives a strong indication of the signal information available through the network nodes: longer STMs indicate more informative measurements. In addition to providing bounds for finite and sparse input sequences, this work also addresses infinite-length input sequences as well as input statistics based on low-rank correlations between multiple input sequences.

In Chapter 4 we address the second research question, which deals with how we can infer signals that are both sparse and temporally correlated as per a dynamic process. Specifically, in Chapter 4 we derive a number of different algorithms that can leverage sparsity and dynamic correlations in a single inference procedure. Initially, it would seem that classical

dynamic filtering algorithms conflicts on a number of levels with the optimization programs popular for sparse signal inference. In particular, dynamic structures are typically used for low-dimensional signals with known and approximately linear dynamics. Alternatively, sparse signal assumptions are most efficient for high-dimensional signals and the dynamics between sparse signals can be complex and nonlinear. Despite these seeming incompatibilities, we derive three methods to combine both sets of information. In keeping with the concepts of measurement efficiency, we first present a simple, computationally-efficient algorithm and provide convergence guarantees to assist with algorithmic parameter selection. To increase robustness and accuracy over this initial algorithm, we also present two more advanced algorithms. One of these algorithms leverages probabilistic hierarchical models to reduce sensitivity to model mismatches. The second algorithms instead learns a parameterized dynamics functions that can increase prediction accuracy. Both of these more advanced algorithms can increase tracking accuracy at an increased computational cost.

The work in Chapter 4, while derived for dynamically correlated signals, can actually be generalized to the significantly broader class of spatially correlated signals. In Chapter 5 we expand our hierarchical dynamic filtering to a general stochastic filtering technique. Specifically, we look at the application of interpreting hyperspectral imagery. In this chapter we discuss the importance of hyperspectral imagery (HSI), and develop unsupervised a dictionary learning algorithm that can extract material decompositions from spatial-spectral measurements. While we discuss a number of applications of the resulting learned spectral dictionaries, we focus on the problem of spectral super-resolution where limited coarse measurements can be resolved into high-fidelity spectral shapes. We utilize a similar hierarchical algorithm to the dynamic filtering algorithm presented in Chapter 4 in order to leverage the high correlations between pixels in HSI scenes. This hierarchical model demonstrates a high accuracy recovery of HSI data both in simulated and real-data experiments.

Finally, to address the third research question, in Chapter 6 we explore how dynamical systems can solve optimization programs used for sparse signal inference. In particular we discuss dynamically evolving networks that are driven by a set of measurements and evolve over time to the signal estimate. We demonstrate that many optimization programs of interest in the context of sparse signal estimation can be implemented in such a network, including more complex optimization programs such as hierarchical models and group-sparse optimizations programs. Additionally, we demonstrate that the performance of the network that solves the most popular of these optimizations, BPDN (Equation (3)), performs as well in real-data applications as state-of-the-art digital optimization routines. While the network-based optimization discussed in Chapter 6 was initially designed to dynamically solve a static optimization program, we also demonstrate a variation of this network which can causally act on streaming signals, such as audio.

CHAPTER II

BACKGROUND

2.1 Sparse Signals and Compressive Sensing

One signal model that has greatly increased signal acquisition efficiency assumes that our signal has many fewer effective degrees of freedom, S, than its ambient dimension, N. In particular, we can say that our data vector $\boldsymbol{x} \in \mathbb{R}^N$ can be described via a generative model

$$\boldsymbol{x} = \boldsymbol{\Psi a} \tag{4}$$

where $\boldsymbol{a} \in \mathbb{R}^{N_2}$ is the sparse vector of coefficients and $\boldsymbol{\Psi} \in \mathbb{R}^{N \times N_2}$ is the matrix whose columns are used in sparse combinations to produce the data vector $\boldsymbol{x}$. This type of model is referred to as a *generative* or *synthesis model*, and $\boldsymbol{a}$ is assumed to have at most S non-zero elements where $S \ll N$. While *analysis models* (i.e. $\boldsymbol{a} = \boldsymbol{\Psi x}$ is assumed to have at most S non-zero elements) also exists, in this work we will focus on the generative model. As a note, the analysis and generative models are identical when $\boldsymbol{\Psi}$ is a basis of ortho-normal vectors.

This sparsity assumption on our signal allows us to infer signals with higher accuracy from many fewer measurements [31, 49, 50]. In particular, *compressive sensing* has focused on deriving very efficient methods to model, sense, and infer sparse signals [51–56]. In compressive sensing, the measurement system, cost functions, and optimization procedures all play a role in the acquisition efficiency. In terms of the measurement process, random projections are shown to capture the signal information robustly, which allows for performance guarantees on convex optimization programs that can be efficiently solved using many standard optimization tools. The resulting theory of CS can robustly estimate sparse signals from many fewer measurements than the signal's ambient dimension.

An important part of compressive sensing involves demonstrating that certain measurement systems can conserve the information in sparse signals the measurement space. One commonly used property that quantifies to what degree a measurement scheme conserves information from sparse signals is the restricted isometry property (RIP). The RIP quantifies, via a parameter δ, how well distances between any two sparse vectors are preserved when observed through the linear measurement system $\mathbf{\Phi}$. Specifically, we say that $\mathbf{\Phi}$ is RIP$(2S, \delta)$ if for all $2S$-sparse signals

$$(1 - \delta) \|\boldsymbol{a}_{2S}\|_2^2 \leq \|\mathbf{\Phi}\mathbf{\Psi}\boldsymbol{a}_{2S}\|_2^2 \leq (1 + \delta) \|\boldsymbol{a}_{2S}\|_2^2 . \tag{5}$$

Smaller δ values indicate that the distances are preserved more stringently and different sparse signals are distinguishable from one another. Figure 2 depicts the benefits of RIP, where 'good' measurements preserve distances between any two sparse signals while 'bad' measurements allow different signals to project onto arbitrarily close points. Typically, showing that a system has the RIP is difficult; however, for randomly generated $\mathbf{\Phi}$ the RIP can be shown to hold with high probability. For example, random Gaussian sensing matrices ($\mathbf{\Phi}$ is a random Gaussian matrix) satisfies the RIP with probability $1 - O(N_2^{-1})$ when M scales linearly with the sparsity S and logarithmically with the representation's ambient dimension N_2. Since $S \ll N$ for many cases, $M < N$ and our signal is recoverable from many fewer measurements than are needed in classical Shannon-Nyquist sampling.

Although the RIP and other, similar, properties show how well a system measures sparse signals, signal recovery still depends on the choice in cost function to optimize. While different philosophies advocate different recovery methods to estimate a sparse signal from a set of measurements, in this work we focus on the Bayesian maximum *a posteriori* (MAP) framework. Generally, MAP estimation blends our observations with assumed prior knowledge by optimizing over probability distributions that represent our relative confidence in $\boldsymbol{y}$ and our expectations for $\boldsymbol{x}$. Specifically, the MAP estimate seeks the signal which maximizes the probability of the signal given the measurements. Equivalently, by Bayes' theorem we can maximize the product of the likelihood of our measurements given

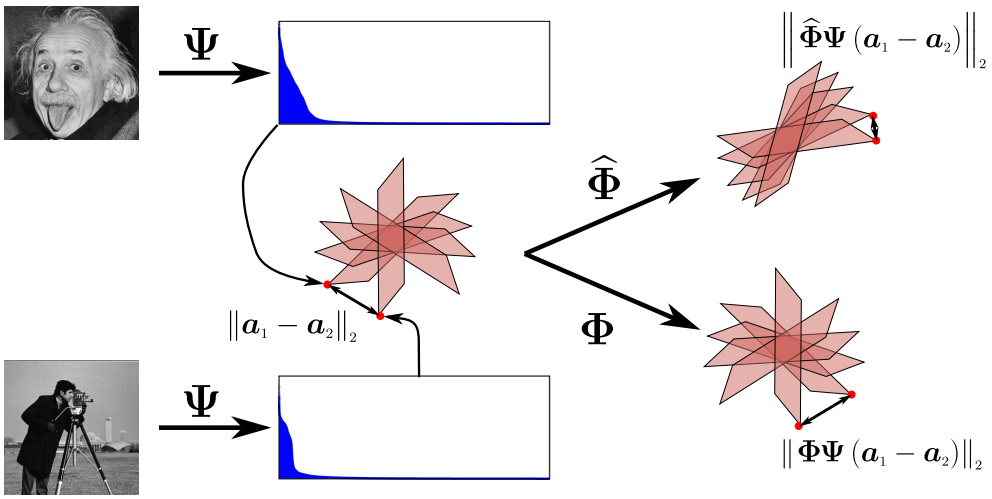

Figure 2: In the compressive sensing framework, we depend on random matrices to provide us with well behaved measurements. In particular, we need the property where different S-sparse signals are still distinguishable after applying the measurement operator $\boldsymbol{\Phi}$. For example two images have different, sparse, wavelet decompositions, as depicted as being two distinct points on a union of S-dimension subspaces. Any measurement operator to be used for compressive measurements should retain relative distances, essentially not allowing these two points to become arbitrarily close after the application of $\boldsymbol{\Phi}$.

the signal and the assumed prior distribution,

$$\widehat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{y}) = \arg\max_{\boldsymbol{x}} p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}). \tag{6}$$

Since many common distributions are in the exponential family, the form

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \left[-\log\left(p(\boldsymbol{y}|\boldsymbol{x})\right) - \log\left(p(\boldsymbol{x})\right)\right], \tag{7}$$

is often used. In this work, we assume that the measurement noise $\boldsymbol{\epsilon}$ is a Gaussian random error with zero mean and covariance matrix $\sigma_\epsilon^2 \boldsymbol{I}_M$,

$$p(\boldsymbol{\epsilon}) = \frac{1}{(2\sigma_\epsilon^2)^{M/2}} e^{-\|\boldsymbol{\epsilon}\|_2^2 / 1\sigma_\epsilon^2}.$$

This noise distribution leads to a Gaussian likelihood function with mean $\boldsymbol{\Phi}\boldsymbol{x}$ and the same covariance matrix, i.e.,

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{(2\sigma_\epsilon^2)^{M/2}} e^{-\|\boldsymbol{y}-\boldsymbol{\Phi}\boldsymbol{x}\|_2^2 / 1\sigma_\epsilon^2}.$$

In this case, the first term of Equation (7) simply becomes the ℓ_2 norm of $\boldsymbol{y} - \boldsymbol{\Phi x}$, and the MAP estimate becomes a regularized least-squares estimate:

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{\Phi x}\|_2^2 + \gamma C(\boldsymbol{x}), \tag{8}$$

where the functional is multiplied through by $2\sigma_\epsilon^2$ to simplify the first term and the regulization function $\gamma C(\boldsymbol{x}) = -2\sigma_\epsilon^2 \log(p(\boldsymbol{x}))$ is the scaled logarithm of the prior distribution[1]

.

Of course, the MAP estimate quality depends heavily on the quality of the signal prior. With no prior information, we ignore the prior term $C(\boldsymbol{x})$ and our estimate reduces to a maximum likelihood (ML) estimate (a least-squares estimate). More often, however, the signal statistics are known to some degree. Here we are interested in priors that reflect our knowledge that $\boldsymbol{x}$ is S-compressible with respect to some (potentially over-complete) dictionary $\boldsymbol{\Psi}$. Distributions which encourage sparsity often have heavier tails than Gaussian distributions and are more tightly peaked around the origin. In Bayesian terms, we can say that these priors have *high kurtotsis*. Many different distributions have high kurtosis and can induce sparse estimates. For example, in [57], the Cauchy distribution,

$$p(\boldsymbol{a}) = \prod_{i=1}^{N_2} \frac{1}{\pi(1 + \boldsymbol{a}[i]^2)},$$

is used as a prior. While having very high kurtosis, this distribution does not result in a convex optimization, and thus can produce inefficiencies in the actual calculation of the signal estimate.

Currently, one of the most widely used distributions for sparse signals is the Laplacian distribution:

$$p(\boldsymbol{a}) = \left(\frac{\lambda}{2}\right)^{N_2} e^{-\lambda\|\boldsymbol{a}\|_1}.$$

MAP estimation under Gaussian measurements and Laplacian priors results in the basis pursuit de-noising (BPDN) cost function (Equation (3)), which is a convex optimization

[1]The variable γ is specifically extracted to highlight its role in trading off between the least-squares measurement fidelity term and cost from the prior distribution

program that can be solved very efficiently in high dimensions. One of the main results in CS blends the measurement and prior quality to show that if the RIP holds for a given $\mathbf{\Phi}$, the sparse representation can be provably recovered via BPDN up to a recovery error given by

$$\|\widehat{\boldsymbol{a}} - \boldsymbol{a}\|_2 \leq C_1 \|\boldsymbol{\epsilon}\|_2 + C_2 \frac{\|\boldsymbol{a} - \boldsymbol{a}_S\|_1}{\sqrt{S}}, \tag{9}$$

where C_1 and C_2 are constants that depend on the RIP constant δ, and $\boldsymbol{a}_S$ is the best S-term approximation to $\boldsymbol{a}$ (the vector composed of the S largest components of $\boldsymbol{a}$ and zero elsewhere). An interesting aspect of the bound in Equation (9) is that it essentially depends on two parts: a measurement quality term dependent on the energy of the measurement error, and a model fit term which depends on the ℓ_1 energy of the signal off of the main support. We note here that many alternate estimation procedures to BPDN exist for estimating a sparse signal from a set of linear measurements, including greedy algorithms [58–61] and alternate optimization programs such as the Dantzig selector [62]. Additionally, alternative methods that do not use the RIP exist for proving accuracy bounds on BPDN [63]. In this work, however, we focus on the BPDN-style optimization programs and the RIP as a measurement quality measure.

2.2 Re-weighted ℓ_1

While theoretical guarantees exist for the Laplacian prior model through analysis of the BPDN optimization, other algorithms can produce empirically superior estimates. For example, priors with higher kurtosis can enforce sparsity more stringently. One such class of distributions we can consider result in ℓ_p norms (with $p < 1$) replacing the ℓ_1 norm in Equation (3). These distributions, as with the Cauchy distribution, result in non-convex optimization problems. Another method to increase estimate accuracy is to make the model more flexible. One way to accomplish this flexibility is to consider a set of hyperparameters $\boldsymbol{\lambda}$ that control relative signal-to-noise ratios between the different coefficients in

a modified BPDN optimization [64]. We can model each individual coefficient as

$$p(\boldsymbol{a}[i]|\boldsymbol{\lambda}[i]) = \frac{\lambda_0 \boldsymbol{\lambda}[i]}{2} e^{-\lambda_0 \lambda[i]|\boldsymbol{a}[i]|}. \tag{10}$$

where λ_0 is a baseline SNR. Since we are only using the $\boldsymbol{\lambda}$ values to introduce flexibility, and we not necessarily interested in inferring $\boldsymbol{\lambda}$, the prior that we actually wish to optimize over is

$$p(\boldsymbol{a}) = \int_{\lambda} p(\boldsymbol{a}|\boldsymbol{\lambda}) p(\boldsymbol{\lambda}) \mathrm{d}\boldsymbol{\lambda},$$

which can have much higher kurtosis than the Laplacian distribution. The advantage to describing the model via the hierarchical structure is that while the actual objective may not be convex, the objective when conditioning on the hyper-parameters can be convex and efficiently solved. Many variational methods have been designed for these types of situations where MAP estimates conditioned on one set of variables is easier to solve than the full MAP estimate [65–67]. Of these methods, one popular algorithm is the expectation-maximization (EM) algorithm. The EM algorithm was designed for scenarios where the marginal distributions of one set of variables with respect to the other set of variables is easily calculated, in addition to the conditional MAP estimate being easily evaluated.

The EM algorithm iteratively refines an estimates of the main variables of interest $\boldsymbol{a}$ and an estimate of the distribution over $\boldsymbol{a}$, as parametrized by $\boldsymbol{\lambda}$, in order to find a local optimal point of the original cost function [68–70]. While there is a rich literature on the EM algorithm, the form we will use here alternates between the following two steps

$$\begin{aligned} &\text{M-Step:} && \widehat{\boldsymbol{a}}^t = \arg\max_{\boldsymbol{a}} p\left(\boldsymbol{y}|\boldsymbol{a}, \boldsymbol{\lambda}^{t-1}\right) p\left(\boldsymbol{a}|\boldsymbol{\lambda}^{t-1}\right) \\ &\text{E-Step:} && \boldsymbol{\lambda}^t = \mathcal{E}_{p\left(\lambda|\widehat{\boldsymbol{a}}^t, \boldsymbol{y}\right)}\left[\boldsymbol{\lambda}\right] \end{aligned}$$

where the algorithmic time t iterates until convergence. In the M-step, we solve the MAP estimate given the conditional prior distribution, using an estimate of the distribution parameters $\boldsymbol{\lambda}$. In the E-step, we find a new conditional prior distribution by finding the expected set of parameters given our new signal estimate. Geometrically, we can think of

each iteration as finding a lower-bound approximation to the non-concave prior in the E-step, then maximizing that surrogate function in the M-step. Since the surrogate function is always less than the actual prior, the iteration can the repeat at the new location, until a local maximum is found. The EM algorithm is particularly useful, since it is guaranteed to converge under mild conditions on the probability distributions [68].

In the context of sparse signal estimation, this EM-style estimation procedure is known as the reweighted-ℓ_1 (RWL1) algorithm. In RWL1 we take the conditional Laplacian priors on $\boldsymbol{a}$ as in Equation (10), and place i.i.d. Gamma distribution with parameters α and θ over the hyper-parameters $\boldsymbol{\lambda}$,

$$p(\boldsymbol{\lambda}[i]) = \frac{\boldsymbol{\lambda}^{\alpha-1}[i]}{\theta^{\alpha}\Gamma(\alpha)} e^{-\boldsymbol{\lambda}_k[i]/\theta},$$

the EM algorithm can be used to iteratively refine the estimates of $\boldsymbol{a}$ and $\boldsymbol{\lambda}$ by alternating between the maximization step (M-step):

$$\widehat{\boldsymbol{a}}^t = \arg\min_{\boldsymbol{a}} \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{a}\|_2^2 + \lambda_0 \|\widehat{\boldsymbol{\Lambda}}^{t-1}\boldsymbol{a}\|_1$$

where $\widehat{\Lambda}^{t-1} = \mathrm{diag}(\widehat{\boldsymbol{\lambda}}^{t-1})$ is a weighting matrix based on the previous estimate of the hyper-parameters, and the expectation step (E-step):

$$\widehat{\boldsymbol{\lambda}}^t[i] = \frac{\kappa}{|\widehat{\boldsymbol{a}}^t[i]| + \beta}$$

where κ and β are constants depending on α and θ, and t indicates algorithmic time (i.e., RWL1 iterates over t until some convergence criteria is met). Computationally, the M-step is a convex optimization that has essentially the same complexity as BPDN, and the E-step is analytic and requires minimal computation, meaning that RWL1 has the computational cost of repeated BPDN. We note that the actual prior that is being (locally) optimized can be calculated as

$$p(\boldsymbol{a}) = \int_{\lambda>0} p(\boldsymbol{a}|\boldsymbol{\lambda}) p(\boldsymbol{\lambda}) \mathrm{d}\boldsymbol{\lambda} = \prod_i \left[\frac{\alpha\theta\lambda_0}{2(\theta\lambda_0|\boldsymbol{a}[i]| + 1)^{\alpha+1}} \right],$$

which is a student-t distribution and has much heavier tails than the Laplacian distribution, and the EM algorithm essentially finds local optima of

$$\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{a}\|_2^2 + \kappa \sum_i \log(\eta|\boldsymbol{a}[i]| + 1),$$

where κ and η are parameters that depend on the distribution parameters λ_0, θ, α and σ_ϵ^2.

2.3 Dictionary Learning

Another way to improve sparse signal estimation deals with the particulars of the sparsifying dictionary $\boldsymbol{\Psi}$. While the dictionary is often assumed known (e.g. wavelet bases for images), in other applications it may not be clear what the best basis to describe a class of signals is. If example data is collected, say a representative pool of data vectors $\{\boldsymbol{x}_k\}$, we can consider learning the dictionary $\boldsymbol{\Psi}$ directly from the data. Here we consider a statistical method of learning dictionaries based on maximizing the probability distribution over $\{\boldsymbol{x}_k\}$. In this method we use Equation (4) to write a likelihood probability distribution on $\boldsymbol{x}$. Essentially, we assume that the matrix $\boldsymbol{\Psi}$ applied to the vector $\boldsymbol{a}$ approximates the data to within a Gaussian difference,

$$p(\boldsymbol{x}|\boldsymbol{a}, \boldsymbol{\Psi}) = \frac{1}{(2\pi\sigma_x^2)^{N/2}} e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\Psi}\boldsymbol{a}\|_2^2}{2\sigma_x^2}} \tag{11}$$

where σ_x is the variance of the reconstruction. To regularize the inference procedure, we can place the Laplacian sparsity prior on the coefficients $\boldsymbol{a}$. Using this prior over all data vectors, we see that the joint posterior distribution over all coefficients for all data vectors is

$$\begin{aligned} \prod_k p(\boldsymbol{a}_k|\boldsymbol{x}_k, \boldsymbol{\Psi}) &\propto \prod_k p(\boldsymbol{x}_k|\boldsymbol{a}_k, \boldsymbol{\Psi}) p(\boldsymbol{a}_k) \\ &\propto \prod_k e^{-\frac{\|\boldsymbol{x}_k-\boldsymbol{\Psi}\boldsymbol{a}_k\|_2^2}{2\sigma_x^2}} e^{-\frac{\sqrt{2}}{\sigma_a}\|\boldsymbol{a}_k\|_1} \end{aligned} \tag{12}$$

where σ_a^2 is the variance of the Laplacian distribution on the coefficients. In Equation (12), the constant scaling factors are dropped since they do not effect the arg max of the posterior distribution. Solving the MAP inference problem yields the coefficients $\{\boldsymbol{a}_k\}$, given a

dictionary $\mathbf{\Psi}$.

$$\begin{aligned}
\{\widehat{\boldsymbol{a}}_k\} &= \arg\max_{\{\boldsymbol{a}\}_k} \left(\prod_k p(\boldsymbol{a}_k | \boldsymbol{x}_k, \mathbf{\Psi} \right) \\
&= \arg\max_{\{\boldsymbol{a}_k\}} \prod_k \left(e^{-\frac{\|\boldsymbol{x}_k - \mathbf{\Psi}\boldsymbol{a}_k\|_2^2}{2\sigma_\epsilon^2}} e^{-\frac{\sqrt{2}}{\sigma_a}\|\boldsymbol{a}_k\|_1} \right) \\
&= \arg\min_{\{\boldsymbol{a}_k\}} \sum_k \left(\|\boldsymbol{x}_k - \mathbf{\Psi}\boldsymbol{a}_k\|_2^2 + \gamma \|\boldsymbol{a}_k\|_1 \right) \qquad (13)
\end{aligned}$$

where $\gamma = 2\sqrt{2}\sigma_\epsilon^2/\sigma_a$. Minimizing the cost function in Equation (13) (the negative log of the posterior) with respect to $\boldsymbol{a}$ given $\mathbf{\Psi}$ coincides with solving the BPDN optimization program independently for each exemplar data vector $\boldsymbol{x}_k$. As a computational note, since these optimization programs are all independent, this procedure is *embarrassingly parallel* and can be solved very quickly, making use of parallel processing toolboxes. To optimize the dictionary, however, the same energy function needs to also be minimized with respect to $\mathbf{\Psi}$. In this Bayesian setting, optimizing $\mathbf{\Psi}$ can be viewed as either a maximum likelihood (ML) estimate or another MAP estimate. In the ML version, we wish to find the dictionary that maximizes the probability of the data given the dictionary, $p(\boldsymbol{x}|\mathbf{\Psi})$, which can be equivalently written as

$$\arg\min_{\mathbf{\Psi}} p(\boldsymbol{x}|\mathbf{\Psi}) = \arg\min_{\mathbf{\Psi}} \int_{\mathbb{R}^N} p(\boldsymbol{x}|\boldsymbol{a}, \mathbf{\Psi}) p(\boldsymbol{a}) d\boldsymbol{a}$$

Optimizing over this distribution, with the integral, would require sampling from the posterior, which can be inaccurate and time intensive. In [57], however, Olshausen and Field show that the distribution is tight about the maximum peak $\widehat{\boldsymbol{a}}$, thus the integral can be estimated by finding the MAP estimate of the coefficients, and instead optimizing the likelihood given the MAP coefficient estimate,

$$\int_{\mathbb{R}^N} p(\boldsymbol{x}|\boldsymbol{a}, \mathbf{\Psi}) p(\boldsymbol{a}) d\boldsymbol{a} \approx \langle p(\boldsymbol{x}|\mathbf{\Psi}, \widehat{\boldsymbol{a}}) p(\boldsymbol{a}) \rangle, \qquad (14)$$

where the notation $\langle \cdot \rangle$ here indicates the empirical mean over the exemplar data. To minimize this likelihood, a gradient descent algorithm can be implemented. Given the coefficients, the gradient step with respect to the i^{th} dictionary element (the i^{th} column of $\mathbf{\Psi}$) is

given by

$$\Delta\boldsymbol{\psi}_i \propto \langle \widehat{\boldsymbol{a}}[i](\boldsymbol{x} - \boldsymbol{\Psi}\widehat{\boldsymbol{a}}) \rangle \tag{15}$$

where $\langle \cdot \rangle$ again denotes the average over the sample set of the data. We can now use the MAP estimation for the coefficients given the dictionary and the gradient descent step over the dictionary given the coefficients to describe an iterative dictionary learning algorithm, as described in Algorithm 1.

Algorithm 1 Sparse coding dictionary learning algorithm of [57]

```
Initialize γ, μ, K, ρ
Initialize Ψ as a random Gaussian matrix
repeat
  for k = 1 to K do
    Choose data example x uniformly at random
    {â_k} = arg min_a ||x − Ψa||_2^2 + γ||a||_1
    Δψ_i(k) = a[i] (x − Ψa)
  end for
  ψ_i ← [ψ_i + (μ/K) Σ_k Δψ_i(k)]
  Normalize the columns of Ψ
  μ ← ρμ
until Ψ converges
```

In Algorithm 1 we initialize the dictionary randomly, and set the SNR sparsity trade-off γ, the number of data samples to use for each gradient step K, the learning rate μ and the rate of decay for the learning rate ρ. The decay in the learning rate is necessary to avoid the algorithm oscillating about a local minimum of the energy function. Additionally, we note that an additional step is necessary to re-normalize the dictionary elements after each gradient step. This step is also necessary, since it prevents the algorithm from converging to a trivial solution where the norms of the columns of $\boldsymbol{\Psi}$ are very large, allowing the coefficient magnitudes to be small, thereby vacuously abiding by the sparsity constraint. This inherent bias in the algorithm stems from the approximation in Equation (14) [57].

While this method, and other related methods that seek to optimize similar optimization functions (i.e. KSVD [33]) have demonstrated utility in signal estimation tasks, additional

methods have been devised that place a prior over the dictionary as well, transforming the ML estimate for $\boldsymbol{\Psi}$ into a MAP estimate [71, 72] (this method is similar to methods used to learn dictionaries for manifold transport operators as well [73]),

$$\begin{aligned} p(\boldsymbol{a}, \boldsymbol{\Psi}|\boldsymbol{x}) &= p(\boldsymbol{x}|\boldsymbol{a}, \boldsymbol{\Psi}) p(\boldsymbol{a}|\boldsymbol{\Psi}) p(\boldsymbol{\Psi}) \\ &\propto e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\Psi}\boldsymbol{a}\|_2^2}{2\sigma_x^2}} e^{-\frac{\sqrt{2}}{\sigma_a}\|\boldsymbol{a}\|_1} e^{-\frac{\|\boldsymbol{\Psi}\|_F^2}{2\sigma_\Psi^2}}. \end{aligned}$$

Placing a prior over the dictionary can improve dictionary learning in two ways. First off, it removes the necessity to re-normalize the dictionary elements at each step. Instead, the dictionary norms can be restricted via a prior which prefers dictionary elements with smaller norms. Additionally, placing a prior can alleviate a major detriment to dictionary learning: the need to specify *a priori* how many dictionary elements will be learned. By placing a norm which peaks at dictionary elements with zero-norm, dictionary elements which are not necessary to represent the data will tend to zero. This means that we can initialize a dictionary with more elements which we expect to need, and allow extraneous dictionary elements to be removed through the learning process. One prior which can be used towards these ends is an *i.i.d*. Gaussian random prior over the dictionary elements with zero mean and variance $\sigma_\Psi^2 \boldsymbol{I}$. The MAP inference is then a joint inference problem given by

$$\{\widehat{\boldsymbol{\Psi}}, \widehat{\boldsymbol{a}}\} = \arg\min_{\{\boldsymbol{\Psi}, \boldsymbol{a}\}} \left(\|\boldsymbol{x} - \boldsymbol{\Psi}\boldsymbol{a}\|_2^2 + \gamma \|\boldsymbol{a}\|_1 + \gamma_\Psi \|\boldsymbol{\Psi}\|_F^2 \right),$$

where $\gamma_\Psi = \sigma_x^2/\sigma_\Psi^2$. The learning procedure we can now derive is essentially the same as in Algorithm 1, but with an extra term added to the gradient descent step,

$$\Delta\boldsymbol{\psi}_i \propto \langle \widehat{\boldsymbol{a}}[i](\boldsymbol{x} - \boldsymbol{\Psi}\widehat{\boldsymbol{a}}) - 2\gamma_\Psi \boldsymbol{\psi}_i \rangle,$$

and the re-normalization stem removed.

CHAPTER III

SHORT-TERM MEMORY IN ECHO-STATE NETWORKS[1]

In the compressive-sensing literature, dynamic-signal observation has played an important part in attempting to lower analog-to-digital sampling rates. A majority of the literature has focused on convolving a signal with a random kernel and sampling at a lower rate than Nyquist (e.g., [74]). However, here we are more concerned with a network framework where a streaming signal enters a randomly connected network and the network nodes are measured once, after the stream has fully entered the network. This sampling process is most closely connected to characterizing the short-term memory (STM) of a network, (the longest input sequence length of past input values recoverable from the current node values).

Characterizing the fundamental limits of STM in networked systems is critical to understanding the computational abilities of these networks [75–82]. Fundamental questions in this area include determining the effects on memory capacity of network size, connectivity patterns, and input statistics. Toward these questions, several researchers [75, 83, 84] have recently investigated network models of the form:

$$\boldsymbol{v}[k] = g\left(\boldsymbol{W}\boldsymbol{v}[k-1] + \boldsymbol{z}x[k] + \widetilde{\boldsymbol{\epsilon}}[N]\right), \tag{16}$$

where $\boldsymbol{v} \in \mathbb{R}^M$ are the network states at time k, $\boldsymbol{W} \in \mathbb{R}^{M\times M}$ is the recurrent (feedback) connectivity matrix, $x[k] \in \mathbb{R}$ is the input sequence at time k, $\boldsymbol{z} \in \mathbb{R}^M$ is the projection of the input into the network, $\widetilde{\boldsymbol{\epsilon}}[N]$ is potential network noise, and $g : \mathbb{R}^M \to \mathbb{R}^M$ is a possible element-wise nonlinearity. The general idea is that if $\boldsymbol{W}$ is rich enough (often

[1] The work presented in this chapter was performed in collaboration with Dr. Han Lun Yap (Sections 3.1.1, 3.1.2, 3.1.5, and 3.2) and Dong Yin (Section 3.1.3). ASC and HLY contributed equally to the work in aforementioned sections. ASC developed the initial problem formulation and ran extensive simulations. The full results presented of these sections are available in [1, 3, 5–9]. ASC and DY also contributed equally to the work in Section 3.1.3 with full results to be presented in [2] with preliminary results presented in [4].

taken as random), a single input will reverberate in the network, creating a "memory" of the past input in the current network states. How past inputs can drive the current network to different states, providing information necessary to recover the input history is depicted in Figure 3 .

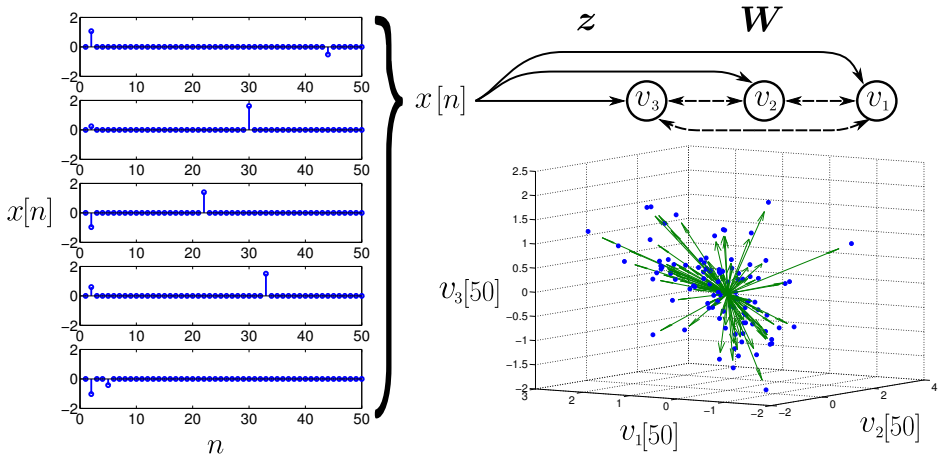

Figure 3: The current state of the network encodes information about the stimulus history. Different stimuli (examples shown to the left), when perturbing the same system (in this figure, a three-neuron-orthogonal network) result in distinct states $\boldsymbol{v} = [v_1, v_2, v_3]^T$ at the current time ($k = 50$). The current state is therefore informative for distinguishing between the input sequences.

The STM capacity of the linear version of this network model (i.e., $g(\boldsymbol{y}) = \boldsymbol{y}$) has been extensively studied [75, 83–85], and as such, the focus is on this network in this proposal. While exact definitions of STM capacity vary, each approach attempts to quantify the amount of information in the current network state available for recovering a past input with some fidelity (e.g., the correlation between the input sequence and the recalled input estimate and Fisher information). These analyses rely on the stochastic nature of the input signal $x[N]$, with [75, 85] specifically assuming Gaussian statistics. These analyses derive STM capacity limits of $N \leq M$, meaning that the number of time samples significantly recoverable by the current network state is limited by the number of nodes in the network. Instead of standard Gaussian models, sparsity models, such as those used in CS, can be used in the context of STM. Using sparsity models for the input statistics, Ganguli

and Sompolinski [84] use an asymptotic statistical mechanics analysis on an approximation of the network dynamics in Equation (16) to argue that orthogonal recurrent network structures can have STM capacities that exceed the number of network nodes. While this work is encouraging, precise bounds based on exact network dynamics would yield deeper insight into the STM of randomly connected networks. Additionally, much of the literature also only addresses STM for single input networks. Multiple input networks are also of interest. These networks evolve similarly as

$$\boldsymbol{v}[k] = g\left(\boldsymbol{W}\boldsymbol{v}[k-1] + \boldsymbol{Z}\boldsymbol{x}[k] + \widetilde{\boldsymbol{\epsilon}}[N]\right), \tag{17}$$

where the input at each time k is now a vector $\boldsymbol{x}[k] \in \mathbb{R}^L$, and the inputs are projected onto the network state by a feed-forward matrix $\boldsymbol{Z} \in \mathbb{R}^{M\times L}$.

Thus, in this chapter we consider dynamics in the observation process. Specifically, as in [75, 83–85], we analyze the STM capacity of linear neural networks that evolve as in Equation (16) with $g(\boldsymbol{v}) = \boldsymbol{v}$. In the STM setting, our measurements are essentially snapshots of the node values[2] at time N ($\boldsymbol{y} = \boldsymbol{v}[N]$) and the signal we wish to recover is the input history, $x[k]$ for $k \in [1, N]$. Our goal is to show that we can invert the process and estimate the perturbations from the node values. As discussed previously, many studies that place no specific model on the inputs indicate that the recoverable input sequence length is bounded by the number of network nodes ($N \leq M$). From a sampling viewpoint, this bound is essentially the Shannon-Nyquist sampling rate. Under sparsity assumptions, however, it appears that we can recover longer sequence ($N \geq M$), indicating that the system is compressively measuring the perturbation sequences [84]. By treating the network as a measurement system, we can show that the network dynamics satisfy the RIP, thereby providing theoretical bounds on the STM for sparse stimuli.

[2]While more generally we can consider reading out the node values in a compressed fashion, i.e., $\boldsymbol{y} = \boldsymbol{C}\boldsymbol{v}[N]$ where $\boldsymbol{C}$ has more columns then rows, here we assume that $\boldsymbol{C} = \boldsymbol{I}$ (i.e., we can read the node values directly) as this assumption isolates the dynamic portion of this problem (information accumulating in $\boldsymbol{v}$).

3.1 STM Capacity using the RIP

3.1.1 Network Dynamics as Compressed Sensing

We consider the same discrete-time ESN model used in previous studies [75, 83–85]:

$$\boldsymbol{y}[k] = f\left(\boldsymbol{W}\boldsymbol{x}[k-1] + \boldsymbol{z}x[k] + \widetilde{\boldsymbol{\epsilon}}[k]\right), \tag{18}$$

where $\boldsymbol{y}[k] \in \mathbb{R}^M$ is the network state at time k, $\boldsymbol{W}$ is the $(M \times M)$ recurrent (feedback) connectivity matrix, $x[k] \in \mathbb{R}$ is the input sequence at time k, $\boldsymbol{z}$ is the $(M \times 1)$ projection of the input into the network, $\widetilde{\boldsymbol{\epsilon}}[k]$ is a potential network noise source, and $f : \mathbb{R}^M \to \mathbb{R}^M$ is a possible pointwise nonlinearity. As in previous studies [75, 83–85], we will consider here the STM capacity of a linear network (i.e., $f(\boldsymbol{y}) = \boldsymbol{y}$).

The recurrent dynamics of Equation (18) can be used to write the network state at time N :

$$\boldsymbol{y}[N] = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{\epsilon}, \tag{19}$$

where $\boldsymbol{\Phi}$ is a $M \times N$ matrix, the k^{th} column of $\boldsymbol{\Phi}$ is $\boldsymbol{W}^{k-1}\boldsymbol{z}$, $\boldsymbol{x} = [x[N], \ldots, x[1]]^T$, the initial state of the system is $\boldsymbol{x}[0] = 0$, and $\boldsymbol{\epsilon}$ is the node activity not accounted for by the input stimulus (e.g. the sum of network noise terms $\boldsymbol{\epsilon} = \sum_{k=1}^{N} \boldsymbol{W}^{N-k}\widetilde{\boldsymbol{\epsilon}}[k]$). With this network model, we assume that the input sequence $\boldsymbol{x}$ is S-sparse in an orthonormal basis $\boldsymbol{\Psi}$ (i.e., there are only S nonzeros in $\boldsymbol{a} = \boldsymbol{\Psi}^T\boldsymbol{x}$).

3.1.2 Single Finite-Length Input

We first consider the STM capacity of a network with single finite-length inputs, where a length-N input signal drives a network and the current state of the M network nodes at time N is used to recover the input history via Equation (3). If $\boldsymbol{\Phi}$ derived from the network dynamics satisfies the RIP for the sparsity basis $\boldsymbol{\Psi}$, the bounds in Equation (9) establish strong guarantees on recovering $\boldsymbol{x}$ from the current network states $\boldsymbol{y}[N]$. Given the significant structure in $\boldsymbol{\Phi}$, it is not immediately clear that any network construction can result in $\boldsymbol{\Phi}$ satisfying the RIP. However, the structure in $\boldsymbol{\Phi}$ is very regular and in fact only

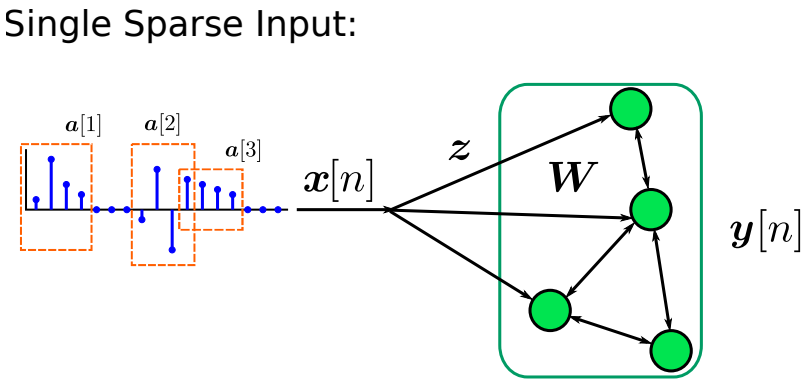

Figure 4: Single input with a sparse structure.

depends on powers of $\boldsymbol{W}$ applied to z:

$$\boldsymbol{\Phi} = \left[z \;\middle|\; \boldsymbol{W}z \;\middle|\; \boldsymbol{W}^2 z \;\middle|\; \ldots \;\middle|\; \boldsymbol{W}^{N-1} z \right].$$

Writing the eigendecomposition of the recurrent matrix $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{-1}$, we re-write the measurement matrix as

$$\boldsymbol{\Phi} = \boldsymbol{U}\left[\tilde{z} \;\middle|\; \boldsymbol{D}\tilde{z} \;\middle|\; \boldsymbol{D}^2\tilde{z} \;\middle|\; \ldots \;\middle|\; \boldsymbol{D}^{N-1}\tilde{z} \right],$$

where $\tilde{z} = \boldsymbol{U}^{-1} z$. Rearranging, we get

$$\boldsymbol{\Phi} = \boldsymbol{U}\widetilde{\boldsymbol{Z}}\left[\boldsymbol{d}^0 \;\middle|\; \boldsymbol{d} \;\middle|\; \boldsymbol{d}^2 \;\middle|\; \ldots \;\middle|\; \boldsymbol{d}^{N-1} \right] = \boldsymbol{U}\widetilde{\boldsymbol{Z}}\boldsymbol{F} \tag{20}$$

where $\boldsymbol{F}_{k,l} = d_k^{l-1}$ is the k^{th} eigenvalue of $\boldsymbol{W}$ raised to the $(l-1)^{\text{th}}$ power and $\widetilde{\boldsymbol{Z}} = \text{diag}\left(\boldsymbol{U}^{-1} z\right)$.

While the RIP conditioning of $\boldsymbol{\Phi}$ depends on all of the matrices in the decomposition of Equation 20, the conditioning of $\boldsymbol{F}$ is the most challenging because it is the only matrix that is compressive (i.e., not square). Due to this difficulty, we start by specifying a network structure for $\boldsymbol{U}$ and $\widetilde{\boldsymbol{Z}}$ that preserves the conditioning properties of $\boldsymbol{F}$ (other network constructions will be discussed in Section 3.2). Specifically, as in [83–85] we choose $\boldsymbol{W}$ to be a random orthonormal matrix, assuring that the eigenvector matrix $\boldsymbol{U}$ has orthonormal columns and preserves the conditioning properties of $\boldsymbol{F}$. Likewise, we choose the feed-forward vector z to be $z = \frac{1}{\sqrt{M}}\boldsymbol{U}\boldsymbol{1}_M$, where $\boldsymbol{1}_M$ is a vector of M ones (the constant $\sqrt{M}$

simplifies the proofs but has no bearing on the result). This choice for z assures that $\widetilde{\boldsymbol{Z}}$ is the identity matrix scaled by $\sqrt{M}$ (analogous to [83] where z is optimized to maximize the SNR in the system). Finally, we observe that the richest information preservation apparently arises for a real-valued $\boldsymbol{W}$ when its eigenvalues are complex, distinct in phase, have unit magnitude, and appear in complex conjugate pairs.

For the above network construction, our main result shows that $\boldsymbol{\Phi}$ satisfies the RIP in the basis $\boldsymbol{\Psi}$ (implying the bounds from Equation (9) hold) when the network size scales linearly with the sparsity level of the input. This result is made precise in the following theorem:

Theorem 1. *Suppose $N \geq M$, $N \geq S$ and $N \geq O(1)$.*[3] *Let $\boldsymbol{U}$ be any unitary matrix of eigenvectors (containing complex conjugate pairs) and set $\boldsymbol{z} = \frac{1}{\sqrt{M}}\boldsymbol{U}\boldsymbol{1}_M$ so that $\widetilde{\boldsymbol{Z}} = diag\left(\boldsymbol{U}^{-1}\boldsymbol{z}\right) = \frac{1}{\sqrt{M}}\boldsymbol{I}$. For M an even integer, denote the eigenvalues of $\boldsymbol{W}$ by $\{e^{jw_m}\}_{m=1}^{M}$. Let the first $M/2$ eigenvalues $\left(\{e^{jw_m}\}_{m=1}^{M/2}\right)$ be chosen uniformly at random on the complex unit circle (i.e., we chose $\{w_m\}_{m=1}^{M/2}$ uniformly at random from $[0, 2\pi)$) and the other $M/2$ eigenvalues as the complex conjugates of these values (i.e., for $M/2 < m \leq M$, $e^{jw_m} = e^{-jw_{m-M/2}}$). Under these conditions, for a given RIP conditioning $\delta < 1$ and failure probability η, if*

$$M \geq C\frac{S}{\delta^2}\mu^2\left(\boldsymbol{\Psi}\right)\log^4\left(N\right)\log(\eta^{-1}), \tag{21}$$

for a universal constant C, then for any $\boldsymbol{x}$ that is S-sparse (i.e., has no more than S non-zero entries)

$$(1-\delta) \leq \|\boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{x}\|_2^2 \,/\, \|\boldsymbol{x}\|_2^2 \leq (1+\delta)$$

with probability exceeding $1-\eta$.

The proof of this statement is given in Appendix 8.4 and follows closely the approach in [53] by generalizing it to both include any basis $\boldsymbol{\Psi}$ and account for the fact that $\boldsymbol{W}$ is a

[3] The notation $N \geq O(1)$ means that $N \geq C$ for some constant C. For clarity, we do not keep track of the constants in our proofs. The interested reader is referred to [53] for specific values of the constants.

real-valued matrix. The quantity $\mu(\cdot)$ (known as the coherence) captures the largest inner product between the sparsity basis and the Fourier basis, and is calculated as:

$$\mu(\mathbf{\Psi}) = \max_{n=1,\ldots,N} \sup_{t\in[0,2\pi]} \left| \sum_{m=0}^{N-1} \mathbf{\Psi}_{m,n} e^{-jtm} \right|. \tag{22}$$

In the result above, the coherence is lower (therefore the STM capacity is higher) when the sparsity basis is more "different" from the Fourier basis.

The main observation of the result above is that STM capacity scales superlinearly with network size. Indeed, for some values of S and $\mu(\mathbf{\Psi})$ it is possible to have STM capacities much greater than the number of nodes (i.e., $N \gg M$). To illustrate the perfect recovery of signal lengths beyond the network size, Figure 5 shows an example recovery of a single long input sequence. Specifically, we generate a 100 node random orthogonal connectivity matrix $\boldsymbol{W}$ and generate $\boldsymbol{z} = \frac{1}{\sqrt{M}}\boldsymbol{U}\mathbf{1}_M$. We then drive the network with an input sequence that is 480 samples long and constructed using 24 non-zero coefficients (chosen uniformly at random) of a wavelet basis. The values at the non-zero entries were chosen uniformly in the range [0.5,1.5]. In this example we omit noise so that we can illustrate the noiseless recovery. At the end of the input sequence, the resulting 100 network states are used to solve the optimization problem in Equation 3 for recovering the input sequence (using the network architecture in [35]). The recovered sequence, as depicted in Figure 5, is identical to the input sequence, clearly indicating that the 100 nodes were able to store the 480 samples of the input sequence (achieving STM capacity higher than the network size).

Directly checking the RIP condition for specific matrices is NP-hard (one would need to check every possible $2S$-sparse signal). In light of this difficulty in verifying recovery of all possible sparse signals (which the RIP implies), we will explore the qualitative behavior of the RIP bounds above by examining in Figure 6 the average recovery relative MSE (rMSE) in simulation for a network with M nodes when recovering input sequences of length N with varying sparsity bases. Figure 6 uses a plotting style similar to the Donoho-Tanner phase transition diagrams [54] where the average recovery rMSE is shown for each pair of variables under noisy conditions. While the traditional Donoho-Tanner phase transitions

plot noiseless recovery performance to observe the threshold between perfect and imperfect recovery, here we also add noise to illustrate the stability of the recovery guarantees. The noise is generated as random additive Gaussian noise at the input ($\widetilde{\boldsymbol{\epsilon}}$ in Equation (18)) to the system with zero mean and variance such that the total noise in the system ($\boldsymbol{\epsilon}$ in Equation (19)) has a norm of approximately 0.01. To demonstrate the behavior of the system, the phase diagrams in Figure 6 sweep the ratio of measurements to the total signal length (M/N) and the ratio of the signal sparsity to the number of measurements (S/M). Thus at the upper left hand corner, the system is recovering a dense signal from almost no measurements (which should almost certainly yield poor results) and at the right hand edge of the plots the system is recovering a signal from a full set of measurements (enough to recover the signal well for all sparsity ranges). We generate ten random ESNs for each combination of ratios (M/N, S/M). The simulated networks are driven with input sequences that are sparse in one of four different bases (Canonical, Daubechies-10 wavelet, Symlet-3 wavelet and DCT) which have varying coherence with the Fourier basis. We use the node values at the end of the sequence to recover the inputs.[4]

In each plot of Figure 6, the dashed line denotes the boundary where the system is able to essentially perform perfect recovery (recovery error $\leq 1\%$) up to the noise floor. Note that the area under this line (the white area in the plot) denotes the region where the system is leveraging the sparse structure of the input to get capacities of $N > M$. We also observe that the dependence of the RIP bound on the coherence with the Fourier basis is clearly shown qualitatively in these plots, with the DCT sparsity basis showing much worse performance than the other bases.

While this first proof was dependent on the deterministic construction for $\boldsymbol{z}$ based on the eigenvectors of $\boldsymbol{W}$, there has also been interest in choosing $\boldsymbol{z}$ as i.i.d. random Gaussian values [83, 84]. In this case, it is also possible to show that $\boldsymbol{\Phi}$ satisfies the RIP (with respect to the basis $\boldsymbol{\Psi}$ and with the same RIP conditioning δ as before) by paying an extra

[4]For computational efficiency, we use the TFOCS software package [86] to solve the optimization problem in Equation (3) for these simulations.

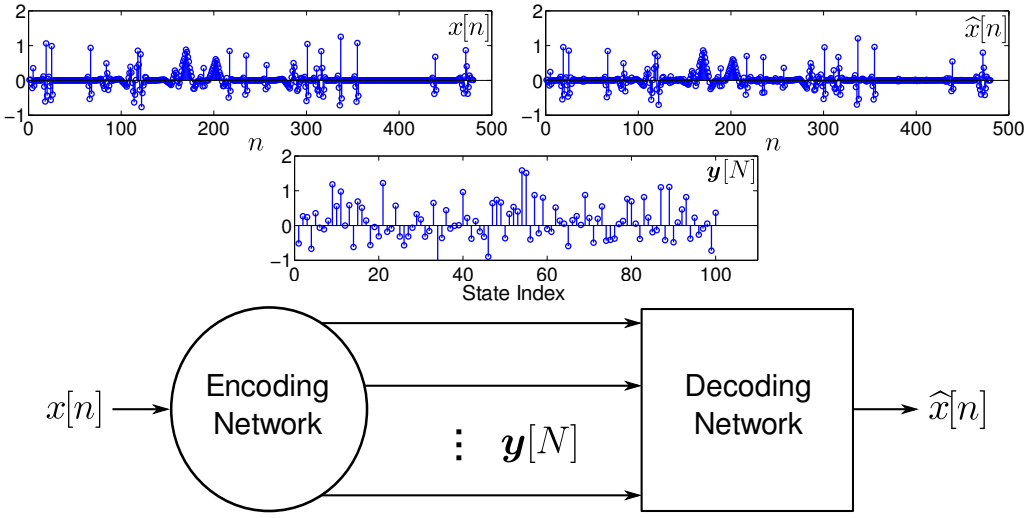

Figure 5: A length 480 stimulus pattern (left plot) that is sparse in a wavelet basis drives the encoding network defined by a random orthogonal matrix $\boldsymbol{W}$ and a feed-forward vector z. The 100 node values (center plot) are then used to recover the full stimulus pattern (right plot) using a decoding network which solves Equation (3).

$\log(N)$ penalty in the number of measurements. Specifically, we have also established the following theorem:

Theorem 2. *Suppose $N \geq M$, $N \geq S$ and $N \geq O(1)$. Let $\boldsymbol{U}$ be any unitary matrix of eigenvectors (containing complex conjugate pairs) and the entries of $\boldsymbol{z}$ be i.i.d. zero-mean Gaussian random variables with variance $\frac{1}{M}$. For M an even integer, denote the eigenvalues of $\boldsymbol{W}$ by $\{e^{jw_m}\}_{m=1}^{M}$. Let the first $M/2$ eigenvalues ($\{e^{jw_m}\}_{m=1}^{M/2}$) be chosen uniformly at random on the complex unit circle (i.e., we chose $\{w_m\}_{m=1}^{M/2}$ uniformly at random from $[0, 2\pi)$) and the other $M/2$ eigenvalues as the complex conjugates of these values. Then, for a given RIP conditioning δ and failure probability $N^{-\log^4 N} \leq \eta \leq \frac{1}{e}$, if*

$$M \geq C \frac{S}{\delta^2} \mu^2 (\boldsymbol{\Psi}) \log^5 (N) \log(\eta^{-1}), \tag{23}$$

$\boldsymbol{\Phi}$ satisfies RIP-(S, δ) with probability exceeding $1 - \eta$ for a universal constant C.

The proof of this theorem can be found in Appendix 8.5. The additional log factor in the bound in (23) reflects that a random feed-forward vector may not optimally spread the input energy over the different eigen-directions of the system. Thus, some nodes may see less

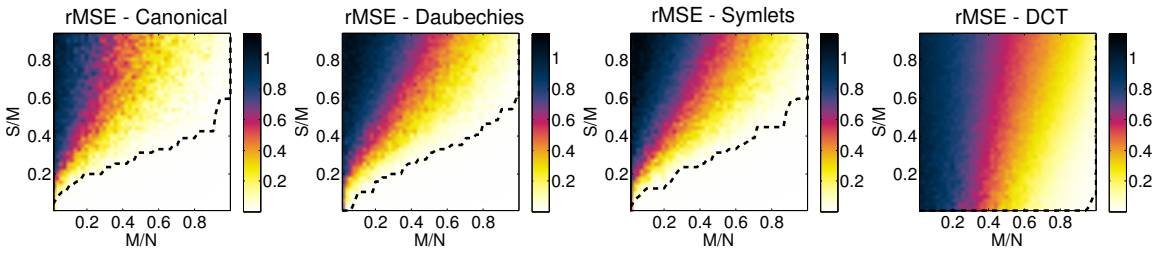

Figure 6: Random orthogonal networks can have a STM capacity that exceeds the number of nodes. These plots depict the recovery relative mean square error (rMSE) for length-1000 input sequences from M network nodes where the input sequences are S-sparse. Each figure depicts recovery for a given set of ratios M/N and S/M. Recovery is near perfect (rMSE $\leq 1\%$; denoted by the dotted line) for large areas of each plot (to the left of the $N = M$ boundary at the right of each plot) for sequences sparse in the canonical basis or various wavelet basis (shown here are 4 level decompositions in Symlet-3 wavelets and Daubechies-10 wavelets). For bases more coherent with the Fourier basis (e.g., discrete cosine transform-DCT), recovery performance above $N = M$ can suffer significantly. All the recovery here was done for noise such that $\|\boldsymbol{\epsilon}\|_2 \approx 0.01$.

energy than others, making them slightly less informative. Note that while this construction does perform worse that the optimal constructions from Theorem 3.1.3, the STM capacity is still very favorable (i.e., a linear scaling in the sparsity level and logarithmic scaling in the signal length).

3.1.3 Sparse Multiple Finite-Length Inputs

While Theorem and Theorem section dealt with networks where the input was only a single stream of inputs, we can also address network constructions where multiple input streams drive the network simultaneously. Specifically we can consider the input to the network at each time step to be a vector rather than a scalar

$$\boldsymbol{y}[n] = \boldsymbol{W}\boldsymbol{y}[n-1] + \sum_{l=1}^{L} \boldsymbol{z}_l x_l[n] + \widetilde{\boldsymbol{\epsilon}}[n] = \boldsymbol{W}\boldsymbol{y}n - 1 + \boldsymbol{Z}\boldsymbol{x}[n] + \widetilde{\boldsymbol{\epsilon}}[n], \tag{24}$$

where now $\boldsymbol{x}[n] \in \mathbb{R}^L$ for all n and $\boldsymbol{Z} \in \mathbb{R}^{M \times L}$ is now a feed-forward matrix, which is composed of concatenating all the individual feed-forward vectors $\boldsymbol{z}_l$. We can analyze this network as with the single input stream network by iterating Equation (24) in on itself:

$$\boldsymbol{y}N = \sum_{k=1}^{N} \boldsymbol{W}^{N-k}\boldsymbol{Z}\boldsymbol{x}[k].$$

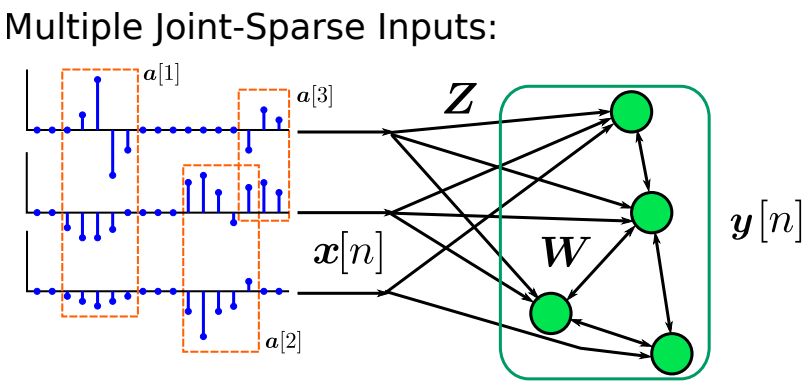

Figure 7: Multiple inputs with a sparse structure.

As in the single input case, we can rewrite the sum as a matrix-vector multiply,

$$\boldsymbol{y}[N] = \left[\boldsymbol{Z}, \boldsymbol{W}\boldsymbol{Z}, \cdots, \boldsymbol{W}^{N-1}\boldsymbol{Z}\right]\left[\boldsymbol{x}^T[N], \boldsymbol{x}^T[N-1], \cdots, \boldsymbol{x}^T[1]\right]^T.$$

and by reorganizing the columns, we can obtain

$$\boldsymbol{y}[N] = \boldsymbol{U}\left[\widetilde{\boldsymbol{Z}}_1\boldsymbol{F}, \widetilde{\boldsymbol{Z}}_2\boldsymbol{F}, \cdots, \widetilde{\boldsymbol{Z}}_L\boldsymbol{F}\right]\left[\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \cdots, \boldsymbol{x}_L^T\right]^T = \boldsymbol{\Phi}\widetilde{\boldsymbol{x}}, \tag{25}$$

where $\boldsymbol{x}_l \in \mathbb{R}^N$ is the l^{th} input stream ($\boldsymbol{x}_l = [x_l[N], \cdots, x_l[1]]^T$) and $\widetilde{\boldsymbol{Z}}_l = \text{diag}(\boldsymbol{U}^{-1}\boldsymbol{z}_l)$ modulates the Fourier measurements for each block ($\boldsymbol{F}$ and $\boldsymbol{U}$ are as described in the single input case). From Equation (26) we can see that the current state is simply the sum of L compressed input streams, where the compression for each block essentially preforms the same compression as the single stream case. While it may be tempting to complete the parallel track to the single input analysis, and to define $\widetilde{\boldsymbol{Z}}$ based on the eigenvectors of $\boldsymbol{W}$, we can quickly see that such a strategy would provide poor results. Specifically, if we choose each $\widetilde{\boldsymbol{Z}}_l$ such that every $\widetilde{\boldsymbol{Z}}_l = \boldsymbol{I}$, then we can see that Equation (26) reduces to

$$\begin{aligned} \boldsymbol{y}N &= \boldsymbol{U}\left[\boldsymbol{F}, \boldsymbol{F}, \cdots, \boldsymbol{F}\right]\left[\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \cdots, \boldsymbol{x}_L^T\right]^T, \\ &= \boldsymbol{U}\boldsymbol{F}\sum_{l=1}^{L}\boldsymbol{x}_l, \end{aligned} \tag{26}$$

which clearly indicates that only the sum of the input streams can be recovered, however the different inputs cannot be distinguished from one another. Instead, we utilize the feed-forward vector style used in Theorem 3.1.3, choosing $\boldsymbol{Z}$ to be a set of random Gaussian

vectors with i.i.d. zero-mean, variance-$1/M$ entries. In this way, each input stream projects differently onto the evolving network state.

Using this setup, we can show a result similar to Theorem 3.1.3, with a few minor modifications. Specifically, the signal model and the resulting coherence term need to be modified to accommodate the new signal input structure. For a single input we could describe the input model as $\boldsymbol{x} = \boldsymbol{\Psi}\boldsymbol{a}$, i.e. $\boldsymbol{x}$ is sparse in $\boldsymbol{\Psi}$. We can similarly describe $\widetilde{bmx} = \boldsymbol{\Psi}\widetilde{\boldsymbol{a}}$, i.e. the composite of all input signals is sparse in a basis $\boldsymbol{\Psi} \in \mathbb{R}^{NL\times NL}$. This means that each signal stream can be written as $\boldsymbol{x}_l = \sum_{k=1}^{L} \boldsymbol{\Psi}^{l,k}\boldsymbol{a}_k$ where $\boldsymbol{\Psi}^{l,k}$ is the $\{l,k\}^{th}$ $NL \times N$ block of $\boldsymbol{\Psi}$. This signal model is very rich in that a given coefficient can influence multiple channels, and the network memory can piece together the interdependencies. With this model, we find it necessary to generalize the coherence parameter used in the previous results.

$$\mu(\boldsymbol{\Psi}) = \max_{l,k=1,\dots,L} \max_{n=1,\dots,N} \sup_{t\in[0,2\pi]} \frac{\left|\sum_{m=0}^{N-1} \boldsymbol{\Psi}^{l,k}_{m,n} e^{-jtm}\right|}{||\boldsymbol{\Psi}^{l,k}_{m}||_2}. \tag{27}$$

In the single input case, the coherence parameter focused on the deviation of the sparsity basis from the Fourier basis. In this multiple input case, each $N\times N$ block must be different from the Fourier basis. This restriction is reasonable, since if a single sub-block of $\boldsymbol{\Psi}$ was coherent with the Fourier basis, then at least one input stream would be sparse in a Fourier-like basis and hence would be unrecoverable. Since we are seeking uniform recovery, this situation is not acceptable. We note that for the case of $L = 1$, the generalized definition of coherence reduces to the definition for single inputs.

Theorem 3. *Suppose $NL \geq M$, $NL \geq S$ and $NL \geq O(1)$. Let $\boldsymbol{U}$ be any unitary matrix of eigenvectors (containing complex conjugate pairs) and the entries of $\boldsymbol{Z}$ be i.i.d. zero-mean Gaussian random variables with variance $\frac{1}{M}$. For M an even integer, denote the eigenvalues of $\boldsymbol{W}$ by $\{e^{jw_m}\}_{m=1}^{M}$. Let the first $M/2$ eigenvalues ($\{e^{jw_m}\}_{m=1}^{M/2}$) be chosen uniformly at random on the complex unit circle (i.e., we chose $\{w_m\}_{m=1}^{M/2}$ uniformly at random from $[0, 2\pi)$) and the other $M/2$ eigenvalues as the complex conjugates of these values. Then, for a given*

RIP conditioning δ and failure probability $N^{-\log^4 N} \leq \eta \leq \frac{1}{e}$, if

$$M \geq C\frac{S}{\delta^2}\mu^2(\mathbf{\Psi})\log^5(NL)\log(\eta^{-1}), \tag{28}$$

where the coherence $\mu(\mathbf{\Phi})$ is defined as in Equation , $\mathbf{\Phi}$ satisfies RIP-(S, δ) with probability exceeding $1 - \eta$ for a universal constant C.

The proof of Theorem 3 is in Appendix 8.6. It is important to notice that when $L = 1$, Theorem 3 reduces to Theorem 3.1.3. As this result is identical in the variables present in the signal input case (S, N, M, etc.), we test this result by testing the dependence on the number of inputs L. Figure 8 depicts the results of a series of simulations where a noiseless set of L signals of fixed temporal length N are fed into a network. We vary the network size, and plot the smallest M where the BPDN optimization program can still perfectly recover the input streams. The resulting relationship between L and M follows a logarithmic curve very closely, and in fact the best fit poly-logarithmic curve only has an exponential of 1.1 (i.e. $M \propto \log^{1.1} L$. This abides by the derived bounds in Theorem 3 and matches the conjectured bounds [55].

3.1.4 Low-Rank Multiple Finite-Length Inputs

In the low rank case, we assume a different type of low dimensional structure. In the sparsity case, the correlations between inputs was defined by the joint dictionary $\mathbf{\Psi}$, wherein one coefficient could describe activity across time and inputs. In the low-rank structure we assume instead that the correlations between nodes arise from a different process. We instead assume that some small number, R, of prototypical signals combine linearly to form the various input streams. Such a signal structure could arise when the input streams come from spatially neighboring locations, and some small number of sources can be observed at each of those locations. In this case, we can write out input matrix in a reduced form

$$\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{V}^*, \tag{29}$$

where $\boldsymbol{V}^* \in \mathbb{R}^{R\times N}$ is the matrix whose rows are the prototypical streams and $\boldsymbol{Q} \in \mathbb{R}^{L\times R}$ represents the mixing matrix that determines how much of each source stream is present

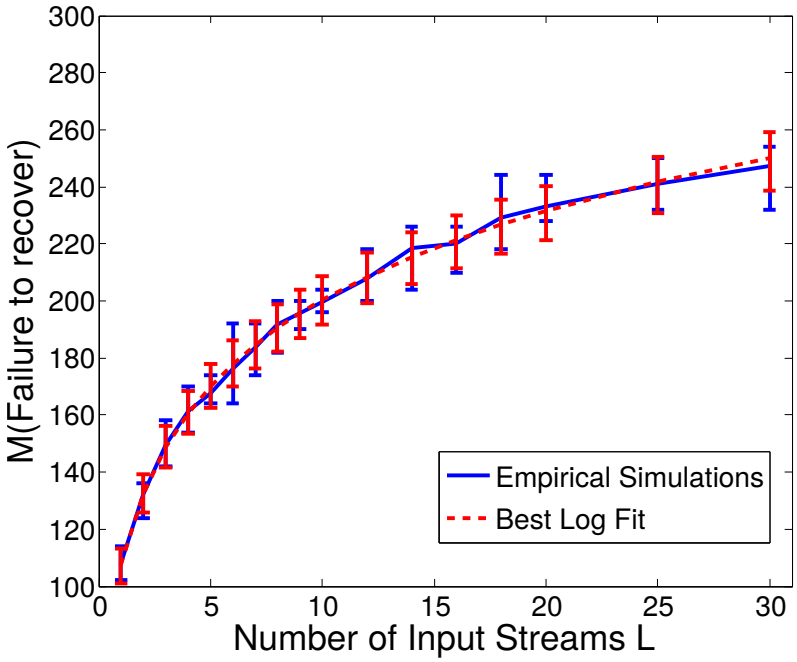

Figure 8: Driving a network with more input sequences has a logarithmic effect on the number of nodes needed to effectively store the inputs driving the system. Empirically, as we increase the number of input streams, the number of nodes needed to recover the signal increases in a logarithmic manner (shown in solid blue). Shown here are the mean M_{failure} over 10 trials, as well as error bars showing the maximum and minimum M_{failure}. The best fit logarithmic function to this curve (and the maximum and minimum values) has an exponent of 1.1 (1.08, and 1.077 for the maximum and minimum respectively).

in each input stream. Since we assume both $L \geq R$ and $N \geq R$, this decomposition of $\boldsymbol{X}$ is a low-rank representation. There is a rich and growing literature dedicated to recovering low-rank matrices from incomplete measurements, the majority focusing on solving the so-called nuclear norm minimization,

$$\min \|\boldsymbol{X}\|_* \quad \text{s.t.} \, \|\boldsymbol{y}[N] - \mathcal{A}(\boldsymbol{X})\|_2 = 0 \tag{30}$$

where the nuclear norm $\|\boldsymbol{X}\|$ is defined as the sum of the singular values of $\boldsymbol{X}$ [87–92]. Nuclear norm minimization is more complex than more standard regularized least-squares optimization programs and initially a more tractable trace-norm minimization was considered [93,94]. Currently, however, proximal methods have made nuclear norm minimization feasible [95–97].

In terms of proving bounds on the solution of Equation (30), while there does exist a

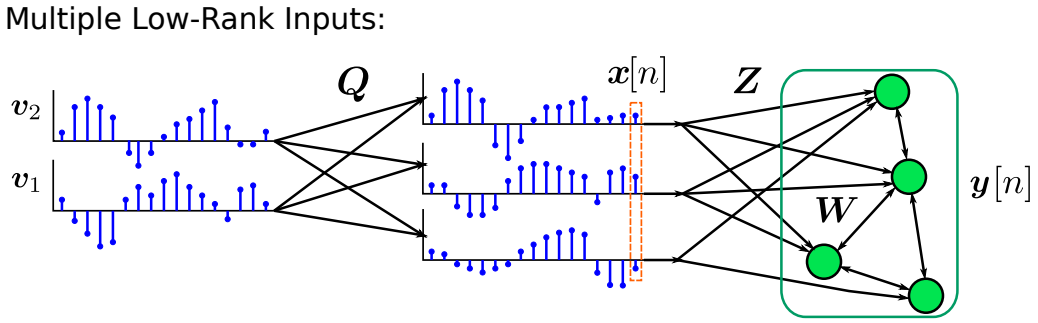

Figure 9: Multiple inputs with a low-rank structure.

comparable property to the RIP for linear operators acting on low-rank matrices, showing the so-called matrix RIP can be difficult even for simple operators. Instead, much of the work in this field instead is based on using a dual-certificate approach [88, 98]. The dual certificate approach uses a proof by construction to show that a dual certificate $\boldsymbol{Y}$ exists, where the projections of $\boldsymbol{Y}$ into and out of the space spanned by the singular vectors of $\boldsymbol{X}$ is bounded appropriately. Specifically we consider the singular value decomposition of $\boldsymbol{X}$ in Equation (29) and the projection $\mathcal{P}_T$ defined as

$$\mathcal{P}_T(\boldsymbol{W}) = \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{W} + \boldsymbol{W}\boldsymbol{V}\boldsymbol{V}^* - \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{W}\boldsymbol{V}\boldsymbol{V}^*$$

which projects a matrix into the space T spanned by the left and right singular vectors. The conditions for the dual certificate $\boldsymbol{Y}$ are then that $\mathcal{A}$ is injective on T and that $\boldsymbol{Y}$ satisfies

$$\begin{aligned} \left\|\mathcal{P}_T(\boldsymbol{Y}) - \boldsymbol{Q}\boldsymbol{V}^H\right\|_F &\leq \frac{1}{2\sqrt{2}\gamma} \\ \|\mathcal{P}_{T^\perp}(\boldsymbol{Y})\| &\leq \frac{1}{2} \end{aligned}$$

where the projection $\mathcal{P}_{T^\perp}$ is the projection onto the perpendicular space to T,

$$\mathcal{P}_{T^\perp}(\boldsymbol{W}) = (\boldsymbol{I} - \boldsymbol{Q}\boldsymbol{Q}^*)\,\boldsymbol{W}\,(\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^*)$$

Using the dual certificate, it has been shown that, as in the case of BPDN, the solution to nuclear norm minimization has provable bounds on its performance both for noiseless and noisy measurements [88]. Specifically, if the dual certificate exists, Equation (30) (the

noiseless measurement case) exactly recovers the low-rank matrix [87]. In the presence of noise, the dual certificate ensures that the solution to the noisy nuclear norm minimization

$$\min \|\boldsymbol{X}\|_* \quad \text{s.t.} \|\boldsymbol{y}[N] - \mathcal{A}(\boldsymbol{X})\|_2 \leq \epsilon, \tag{31}$$

where $\epsilon = \|\boldsymbol{\epsilon}\|_2$ is the ℓ_2 norm of the measurement error, satisfies the inequality

$$\left\|\widehat{\boldsymbol{X}} - \boldsymbol{X}\right\| \leq \left(4\sqrt{\min(N, L)\frac{2NL + M}{M}} + 2\right)\epsilon. \tag{32}$$

We note that while this bound is looser than desired in the sense that the bound grows with the problem size, alternate optimization routines can demonstrate the desired scaling laws [98]. While these bounds are provable in this case, we present here only the nuclear norm results to retain mathematical consistency.

Here we use the dual certificate tools to derive the following theorem:

Theorem 4. *Suppose $NL \geq M$, $N \geq R$, $N \geq O(1)$ and $L \geq O(1)$. Let $\boldsymbol{z}$ be i.i.d. zero-mean Gaussian random variables with variance $\frac{1}{M}$. For M an even integer, denote the eigenvalues of $\boldsymbol{W}$ by $\{e^{jw_m}\}_{m=1}^{M}$. Let the first $M/2$ eigenvalues ($\{e^{jw_m}\}_{m=1}^{M/2}$) be chosen uniformly at random on the complex unit circle (i.e., we chose $\{w_m\}_{m=1}^{M/2}$ uniformly at random from $[0, 2\pi)$) and the other $M/2$ eigenvalues as the complex conjugates of these values. If the number of nodes scales as*

$$M \geq cR\left(N + \mu_0^2 L\right)\log^3(LN),$$

where the coherence parameter is defined as

$$\mu_0^2 = R \sup_{\omega \in [0, 2\pi]} \left\|\boldsymbol{V}^* \boldsymbol{f}_\omega\right\|_2^2,$$

then, with probability at least $1 - O((LN)^{1-\beta}$ the minimization in Equation (30) *exactly recovers the rank-R input matrix $\boldsymbol{X}$ under noiseless conditions and the minimization* (31) *recovers $\boldsymbol{X}$ to within the error* (32) *under noisy conditions.*

The proof of Theorem 4 is in Appendix 8.7 and follows a traditional glofing scheme to find an inexact dual certificate. In fact, we note that since our architecture is similar

mathematically to the architecture in [98], our proof is similar as well. The main difference is that due to the unbounded nature of our distributions, and the fact that our Fourier vectors are continuously random, rather than gridded, we can consider our proof as a generalization of the proof in [98].

3.1.5 STM Capacity of Infinite-Length Inputs

After establishing the perfect recovery bounds for finite-length inputs in the previous section, we turn here to the more interesting case of a network that has received an input beyond its STM capacity (perhaps infinitely long). In contrast to the finite-length input case where favorable constructions for $\boldsymbol{W}$ used random unit-norm eigenvalues, this construction would be unstable for infinitely long inputs. In this case, we take $\boldsymbol{W}$ to have all eigenvalue magnitudes equal to $q < 1$ to ensure stability. The matrix constructions we consider in this section are otherwise identical to that described in the previous section.

In this scenario, the recurrent application of $\boldsymbol{W}$ in the system dynamics assures that each input perturbation will decay steadily until it has zero effect on the network state. While good for system stability, this decay means that each input will slowly recede into the past until the network activity contains no useable memory of the event. In other words, *any* network with this decay can only hope to recover a proxy signal that accounts for the decay in the signal representation induced by the forgetting factor q. Specifically, we define this proxy signal to be $\boldsymbol{Q}\boldsymbol{x}$, where $\boldsymbol{Q} = \text{diag}\left(\left[1, q, q^2, \ldots\right]\right)$. Previous work [75, 83, 85] has characterized recoverability by using statistical arguments to quantify the correlation of the node values to each past input perturbation. In contrast, our approach is to provide recovery bounds on the rMSE for a network attempting to recover the N past samples of $\boldsymbol{Q}\boldsymbol{x}$, which corresponds to the weighted length-N history of $\boldsymbol{x}$. Note that in contrast to the previous sections where we established the length of the input that can be perfectly recovered, the amount of time we attempt to recall (N) is now a parameter that can be varied.

Our technical approach to this problem comes from observing that activity due to inputs older than N acts as interference when recovering more recent inputs. In other words, we

can group older terms (i.e., from farther back than N time samples ago) with the noise term, resulting again in $\mathbf{\Phi}$ being an M by N linear operation that can satisfy RIP for length-N inputs. In this case, after choosing the length of the memory to recover, the guarantees in Equation (9) hold when considering every input older than N as contributing to the "noise" part of the bound.

Specifically, in the noiseless case where $\boldsymbol{x}$ is sparse in the canonical basis ($\mu(\boldsymbol{I}) = 1$) with a maximum signal value $x_{\max}$, we can bound the first term of Equation (9) using a geometric sum that depends on N, S and q. For a given scenario (i.e., a choice of q, S and the RIP conditioning of $\mathbf{\Phi}$), a network can support signal recovery up to a certain sparsity level S^*, given by:

$$S^* = \frac{M\delta^2}{C\log^\gamma(N)}, \tag{33}$$

where γ is a scaling constant (e.g., $\gamma = 4$ using the present techniques, but $\gamma = 1$ is conjectured [55]). We can also bound the second term of Equation (9) by the sum of the energy in the past N perturbations that are beyond this sparsity level S^*. Together these terms yield the bound on the recovery of the proxy signal:

$$\begin{aligned} \left\|\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{Q}\widehat{\boldsymbol{x}}\right\|_2 \;&\leq\; \beta x_{\max} \|\boldsymbol{U}\|_2 \left(\frac{q^N}{1-q}\right) \\ &+\; \frac{\beta x_{\max}}{\sqrt{\min[S^*, S]}} \left(\frac{q^{\min[S^*,S]} - q^S}{1-q}\right) \\ &+\; \alpha\epsilon_{\max} \|\boldsymbol{U}\|_2 \left|\frac{q}{1-q}\right|. \end{aligned} \tag{34}$$

The derivation of the first two terms in the above bound is detailed in Appendix 8.8, and the final term is simply the accumulated noise, which should have bounded norm due to the exponential decay of the eigenvalues of $\boldsymbol{W}$.

Intuitively, we see that this approach implies the presence of an optimal value for the recovery length N. For example, choosing N too small means that there is useful signal information in the network that the system is not attempting to recover, resulting in omission errors (i.e., an increase in the first term of Equation (9) by counting too much signal

as noise). On the other hand, choosing N too large means that the system is encountering recall errors by trying to recover inputs with little or no residual information remaining in the network activity (i.e., an increase in the second term of Equation (9) from making the signal approximation worse by using the same number of nodes for a longer signal length).

The intuitive argument above can be made precise in the sense that the bound in Equation (34) does have at least one local minimum for some value of $0 < N < \infty$. First, we note that the noise term (i.e., the third term on the right side of Equation (34)) does not depend on N (the choice in origin does not change the infinite summation), implying that the optimal recovery length only depends on the first two terms. We also note the important fact that S^* is non-negative and monotonically decreasing with increasing N. It is straightforward to observe that the bound in equation Equation (34) tends to infinity as N increases (due to the presence of S^* in the denominator of the second term). Furthermore, for small values of N, the second term in Equation (34) is zero (due to $S^* > S$), and the first term is monotonically decreasing with N. Taken together, since the function is continuous in N, has negative slope for small N and tends to infinity for large N, we can conclude that it must have at least one local minima in the range $0 <$N$< \infty$. This result predicts that there is (at least one) optimal value for the recovery length N.

The prediction of an optimal recovery length above is based on the fact that the error bound in Equation (34)), and it is possible that the error itself will not actually show this behavior (since the bound may not be tight in all cases). To test the qualitative intuition from Equation (34), we simulate recovery of input lengths and show the results in Figure 10. Specifically, we generate 50 ESNs with 500 nodes and a decay rate of q=0.999. The input signals are length-8000 sequences that have 400 nonzeros whose locations are chosen uniformly at random and whose amplitudes are chosen from a Gaussian distribution (zero mean and unit variance). After presenting the full 8000 samples of the input signal to the network, we use the network states to recover the input history with varying lengths and compared the resulting MSE to the bound in Equation (34). Note that while the theoretical

bound may not be tight for large signal lengths, the recovery MSE matches the qualitative behavior of the bound by achieving a minimum value at $N > M$.

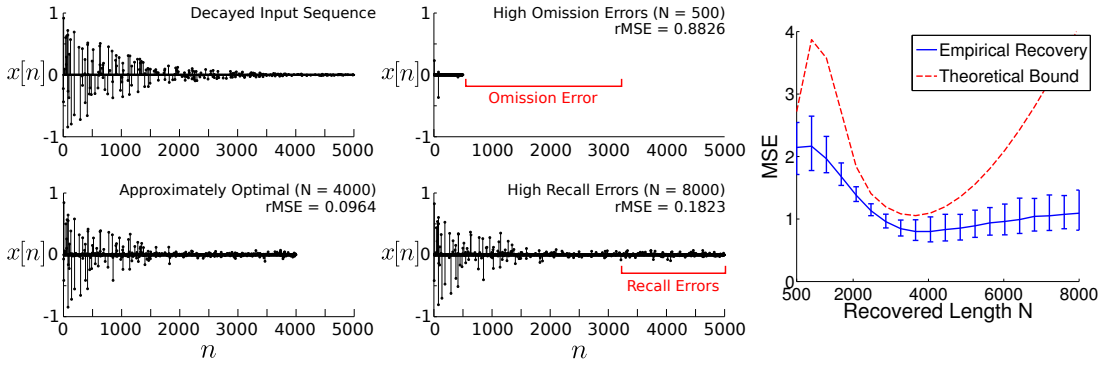

Figure 10: The theoretical bound on the recovery error for the past N perturbations to a network of size M has a minimum value at some optimal recovery length. This optimal value depends on the network size, the sparsity S, the decay rate q, and the RIP conditioning of $\mathbf{\Phi}$. Shown on the right is a simulation depicting the MSE for both the theoretical bound (red dashed line) and an empirical recovery for varying recovery lengths N. In this simulation $S = 400$, $q = 0.999$, $M = 500$. The error bars for the empirical curve show the maximum and minimum MSE. On the left we show recovery of a length-8000 decayed signal (top left) when recovering the past 500 (top right), 4000 (bottom left), and 8000 (bottom right) most recent perturbations. As expected, at $N = 4000$ (approximately optimal) the recovery has the highest accuracy.

3.2 Other Network Constructions

3.2.1 Alternate Orthogonal Constructions

Our results in the previous section focus on the case where $\boldsymbol{W}$ is orthogonal and $\boldsymbol{z}$ projects the signal evenly into all eigenvectors of $\boldsymbol{W}$. When either $\boldsymbol{W}$ or $\boldsymbol{z}$ deviate from this structure the STM capacity of the network apparently decreases. In this section we revisit those specifications, considering alternate network structures allowed under these assumptions as well as the consequences of deviating from these assumptions in favor of other structural advantages for a system (e.g., wire length, etc.).

To begin, we consider the assumption of orthogonal network connectivity, where the eigenvalues have constant magnitude and the eigenvectors are orthonormal. Constructed in this way, $\boldsymbol{U}$ exactly preserves the conditioning of $\widetilde{\boldsymbol{Z}}\boldsymbol{F}$. While this construction may seem

restrictive, orthogonal matrices are relatively simple to generate and encompass a number of distinct cases. For small networks, selecting the eigenvalues uniformly at random from the unit circle (and including their complex conjugates to ensure real connectivity weights) and choosing an orthonormal set of complex conjugate eigenvectors creates precisely these optimal properties. For larger matrices, the connectivity matrix can instead be constructed directly by choosing $\boldsymbol{W}$ at random and orthogonalizing the columns. Previous results on random matrices [99] guarantee that as the size of $\boldsymbol{W}$ increases, the eigenvalue probability density approaches the uniform distribution as desired. Some recent work in STM capacity demonstrates an alternate method by which orthogonal matrices can be constructed while constraining the total connectivity of the network [78]. This method iteratively applies rotation matrices to obtain orthogonal matrices with varying degrees of connectivity. We note here that one special case of connectivity matrices not well-suited to the STM task, even when made orthogonal, are symmetric networks, where the strictly real-valued eigenvalues generates poor RIP conditioning for $\boldsymbol{F}$.

While simple to generate in principle, the matrix constructions discussed above are generally densely connected and may be impractical for many systems. However, many other special network topologies that may be more biophysically realistic (i.e., block diagonal connectivity matrices and small-world[5] networks [100]) can be constructed so that $\boldsymbol{W}$ still has orthonormal columns. For example, consider the case of a block diagonal connection matrix (illustrated in Figure 11), where many unconnected networks of at least two nodes each are driven by the same input stimulus and evolve separately. Such a structure lends itself to a modular framework, where more of these subnetworks can be recruited to recover input stimuli further in the past. In this case, each block can be created independently as above and pieced together. The columns of the block diagonal matrix will still have unit

[5] Small-world structures are typically taken to be networks where small groups of neurons are densely connected amongst themselves, yet sparse connections to other groups reduces the maximum distance between any two nodes.

norm and will be both orthogonal to vectors within its own block (since each of the diagonal sub-matrices are orthonormal) and orthogonal to all columns in other blocks (since there is no overlap in the non-zero indices).

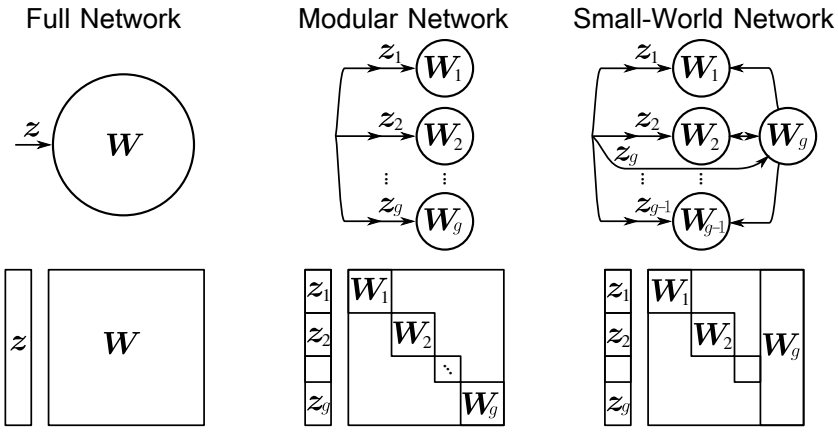

Figure 11: Possible network topologies which have orthogonal connectivity matrices. In the general case, all nodes are connected via non-symmetric connections. Modular topologies can still be orthogonal if each block is itself orthogonal. Small world topologies may also have orthogonal connectivity, especially when a few nodes are completely connected to a series of otherwise disjoint nodes.

Similarly, a small-world topology can be achieved by taking a few of the nodes in every group of the block diagonal case and allowing connections to all other neurons (either unidirectional or bidirectional connections). To construct such a matrix, a block diagonal orthogonal matrix can be taken, a number of columns can be removed and replaced with full columns, and the resulting columns can be made orthonormal with respect to the remaining block-diagonal columns. In these cases, the same eigenvalue distribution and eigenvector properties hold as the fully connected case, resulting in the same RIP guarantees (and therefore the same recovery guarantees) demonstrated earlier. We note that this is only one approach to constructing a network with favorable STM capacity and not all networks with small-world properties will perform well.

Additionally, we note that as opposed to networks analyzed in prior work (in particular the work in [79] demonstrating that random networks with high connectivity have short

STM), the average connectivity does not play a dominant role in our analysis. Specifically, it has been observed in spiking networks that higher network connectivity can reduce the STM capacity so that is scales only with $\log(M)$ [79]). However, in our ESN analysis, networks can have low connectivity (e.g. 2x2 block-diagonal matrices - the extreme case of the block diagonal structure described above) or high connectivity (e.g. fully connected networks) and have the same performance.

3.2.2 Suboptimal Network Constructions

Finally, we can also analyze some variations to the network structure assumed in this paper to see how much performance decreases. For example, instead of orthogonal connectivity matrices, there has also been interest in network constructions involving non-orthogonal connectivity matrices (perhaps for noise reduction purposes [83]). When the eigenvalues of $\boldsymbol{W}$ still lie on the complex unit circle, we can analyze how non-orthogonal matrices affect the RIP results. In this case, the decomposition in Equation (20) still holds and Theorem 3.1.3 still applies to guarantee that $\boldsymbol{F}$ satisfies the RIP. However, the non-orthogonality changes the conditioning of $\boldsymbol{U}$ and subsequently the total conditioning of $\boldsymbol{Phi}$. Specifically the conditioning of $\boldsymbol{U}$ (the ratio of the maximum and minimum singular values $\sigma^2_{\max}/\sigma^2_{\min} = \gamma$) will effect the total conditioning of $\boldsymbol{Phi}$. We can use the RIP of $\boldsymbol{F}$ and the extreme singular values of $\boldsymbol{U}$ to bound how close $\boldsymbol{UF}$ is to an isometry for sparse vectors, both above by

$$\|\boldsymbol{UFx}\|_2^2 \leq \sigma^2_{\max} \|\boldsymbol{Fx}\|_2^2 \leq \sigma^2_{\max} C(1+\delta) \|\boldsymbol{x}\|_2^2 ,$$

and below by

$$\|\boldsymbol{UFx}\|_2^2 \geq \sigma^2_{\min} \|\boldsymbol{Fx}\|_2^2 \geq \sigma^2_{\min} C(1-\delta) \|\boldsymbol{x}\|_2^2 .$$

By consolidating these bounds, we find a new RIP statement for the composite matrix

$$C'(1-\delta') \|\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{UFx}\|_2^2 \leq C'(1+\delta') \|\boldsymbol{x}\|_2^2$$

where $\sigma_{\min}^2 C(1-\delta) = C'(1-\delta')$ and $\sigma_{\max}^2 C(1+\delta) = C'(1+\delta')$. These relationships can be used to solve for the new RIP constants:

$$\begin{aligned} \delta' &= \frac{\frac{\gamma-1}{\gamma+1} + \delta}{1 + \delta\frac{\gamma-1}{\gamma+1}} \\ C' &= \frac{1}{2} C \left(\sigma_{\max}^2 + \sigma_{\min}^2 + \delta(\sigma_{\max}^2 - \sigma_{\min}^2) \right) \end{aligned}$$

These expressions demonstrate that as the conditioning of $\boldsymbol{U}$ improves (i.e. $\gamma \to 1$), the RIP conditioning does not change from the optimal case of an orthogonal network ($\delta' = \delta$). However, as the conditioning of $\boldsymbol{U}$ gets worse and γ grows, the constants associated with the RIP statement also get worse (implying more measurements are likely required to guarantee the same recovery performance).

The above analysis primarily concerns itself with constructions where the eigenvalues of $\boldsymbol{W}$ are still unit norm, however $\boldsymbol{U}$ is not orthogonal. Generally, when the eigenvalues of $\boldsymbol{W}$ differ from unity and are not all of equal magnitude, the current approach becomes intractable. In one case, however, there are theoretical guarantees: when $\boldsymbol{W}$ is rank deficient. If $\boldsymbol{W}$ only has $\widetilde{M}$ unit-norm eigenvalues, and the remaining $M - \widetilde{M}$ eigenvalues are zero, then the resulting matrix $\boldsymbol{\Phi}$ is composed the same way, except that the bottom $M - \widetilde{M}$ rows are all zero. This means that the effective measurements only depend on an $\widetilde{M} \times N$ subsampled DTFT

$$\begin{aligned} \boldsymbol{y}[N] &= \boldsymbol{U}\widetilde{\boldsymbol{Z}}\boldsymbol{F}\boldsymbol{x} + \boldsymbol{\epsilon} \\ &= \boldsymbol{U}\widetilde{\boldsymbol{Z}} \begin{bmatrix} \widetilde{\boldsymbol{F}} \\ \boldsymbol{0}_{M-\widetilde{M},N} \end{bmatrix} \boldsymbol{x} + \boldsymbol{\epsilon} \\ &= \boldsymbol{U}\widetilde{\boldsymbol{Z}}_{1:\widetilde{M}} \widetilde{\boldsymbol{F}}\boldsymbol{x} + \boldsymbol{\epsilon} \end{aligned}$$

where $\widetilde{\boldsymbol{F}}$ is matrix consisting of the non-zero rows of $\boldsymbol{F}$. In this case we can choose any $\widetilde{M}$ of the nodes and the previous theorems will all hold, replacing the true number of nodes M with the effective number of nodes $\widetilde{M}$.

3.3 Discussion

This chapter outlines how the tools from the compressive sensing literature can provide a way to quantify the STM capacity in linear networks using rigorous non-asymptotic recovery error bounds. Of particular note is that this approach leverages the non-Gaussianity of the input statistics to show STM capacities that are super-linear in the size of the network and depend linearly on the sparsity level of the input. This work provides a concrete theoretical understanding for the approach conjectured in [84] along with a generalization to multiple-input networks, arbitrary sparsity bases and infinitely long input sequences. This analysis also predicts that there exists an optimal recovery length that balances omission errors and recall mistakes.

In contrast to previous work on ESNs that leverage nonlinear network computations for computational power [101], the present work uses a linear network and nonlinear computations for signal recovery. Despite the nonlinearity of the recovery process, the fundamental results of the CS literature also guarantee that the recovery process is stable and robust. For example, with access to only a subset of nodes (due to failures or communication constraints), signal recovery generally degrades gracefully by still achieving the best possible approximation of the signal using fewer coefficients. Beyond signal recovery, we also note that the RIP can guarantee performance on many tasks (e.g. detection, classification, etc.) performed directly on the network states [102].

CHAPTER IV

TRACKING OF TIME-VARYING SIGNALS[1]

While the use of sparsity-inducing priors in MAP estimation is discussed in Section 2.1, another historically important prior is based on dynamic information. Many signals of interest are not independent, but instead result from a process that produces many, correlated signals:

$$\boldsymbol{x}_n = f(\boldsymbol{x}_{n-1}) + \boldsymbol{\nu}_n, \tag{35}$$

where n represents the time index, $f : \mathbb{R}^N \to \mathbb{R}^N$ is the dynamics function that represents knowledge of how the time-varying signal $\boldsymbol{x}_n$ evolves, and $\boldsymbol{\nu} \in \mathbb{R}^N$ (called the innovations) represents the modeling error in the dynamics function. The measurements are again taken via linear projections:

$$\boldsymbol{y}_n = \boldsymbol{\Phi}_n \boldsymbol{x}_n + \boldsymbol{\epsilon}_n, \tag{36}$$

where the measurement matrix may differ at each time step. The estimation problem in this setting becomes more complex, as using the temporal information seems to require a joint estimation of all the correlated signals. One canonical result, however, states that for linear $f(\boldsymbol{x}) = \boldsymbol{F}\boldsymbol{x}$ and Gaussian $\boldsymbol{\nu}_n$ and $\boldsymbol{\epsilon}_n$, optimal estimates can be obtained efficiently and causally (that is that $\boldsymbol{x}_n$ can be estimated at time n using only $\boldsymbol{y}_k$ for $k \leq n$). This algorithm, the Kalman filter, essentially propagates a distribution of the estimate $\widehat{\boldsymbol{x}}_n$ forward in time, using $\widehat{\boldsymbol{x}}_n$'s mean and variance with the new measurements to estimate the signal at the next time step [103]. At each time n, the Kalman filter essentially solves

$$\widehat{\boldsymbol{x}}_n = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y}_n - \boldsymbol{\Phi}_n \boldsymbol{x}\|^2_{2,R_n} + \|\boldsymbol{x} - \boldsymbol{F}\widehat{\boldsymbol{x}}_{n-1}\|^2_{2,\boldsymbol{F}\boldsymbol{P}_{n-1}\boldsymbol{F}^T + \boldsymbol{Q}_n},$$

[1]This chapter is in collaboration with Dr. Salman Asif and Dr. Justin Romberg (Section 4.1), and Dr. Aurele Balavoine (Section 4.2.2). ASC was the primary author for the work in 4.1 with more details available in [15]. Full details on other work presented in this section are available in [10–14, 16–19]

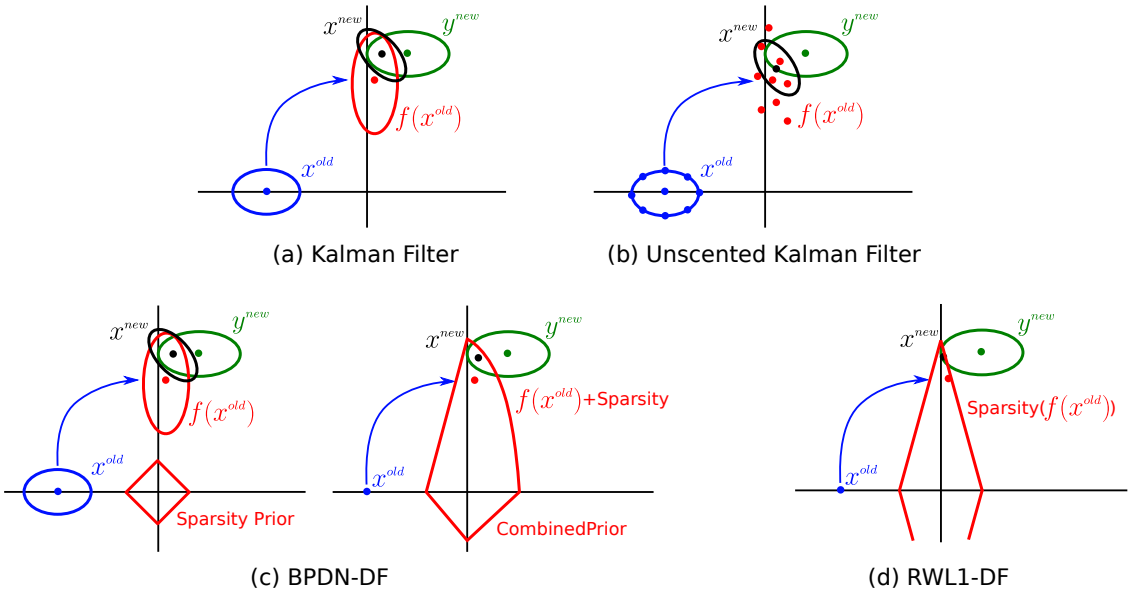

Figure 12: Information propagation in dynamic filtering algorithms. (a) Standard Kalman filtering approaches propagate the mean and covariance to generate the next state's prior distribution. This prior is used in conjunction with the new measurements through the likelihood function to estimate the new state and its distribution. (b) Particle filters estimate the prior distribution via an empirical sampling process that approximates the distribution's moments. (c) Adding the sparsity prior in directly to the Kalman filter optimization (as in BPDN-DF) results in a regularization norm which does not promote sparsity as well as desired. Left: Previous state estimates can still be propagated through the dynamic model to generate a prior that can be combined with an additional prior to encourage sparsity (both in red). Right: Combining the two priors from the left diagrams shows a total signal prior that is curved outward more like an ℓ_2 ball. The convex shape is less effective at promoting sparsity than the ℓ_1 ball. (d) RWL1-DF uses the previous estimate to set the parameters of a prior that has the diamond-like shape known to promote sparse solutions.

where $\boldsymbol{P}_n$, $\boldsymbol{R}_n$ and $\boldsymbol{Q}_n$ are the covariance matrices of $\widehat{\boldsymbol{x}}_n$, $\boldsymbol{\epsilon}_n$ and $\boldsymbol{\nu}_n$, respectively, and the matrix ℓ_2 norm is defined as $\|\boldsymbol{x}\|^2_{2,\boldsymbol{A}} = \boldsymbol{x}^T\boldsymbol{A}^{-1}\boldsymbol{x}$. Geometrically, (depicted in Figure 12(a)) the Kalman filter can be described as projecting the last estimate's distribution forward through the dynamics model, where it is weighted against the measurements by the covariance of $\boldsymbol{\epsilon}_n$. Although this causal estimator is computationally simpler than joint estimation of the states, it still calculates the same estimate for $\widehat{\boldsymbol{x}}_n$ as if all of the previous data had been used.

Unfortunately, the analytic simplicity and optimality guarantees of the Kalman filter

are highly dependent on the linear and Gaussian model assumptions. Although not optimal, many heuristic approaches follow the spirit of the Kalman filter, while incorporating nonlinear system dynamics or non-Gaussian structure. For example, the Extended Kalman Filter [104] incorporates (weakly) nonlinear system dynamics via a linear approximation to $f(\cdot)$. Alternatively, for highly non-linear functions or non-Gaussian statistics, particle filtering uses discrete points (particles) to approximate relevant distributions and propagate those distributions through nonlinear dynamics. The Unscented Kalman filter [105] (see Figure 12(b)) is an example of this technique with a deterministic (rather than typical Monte-Carlo) particles sampling scheme. Though particle filtering approaches do seek the true prior distribution, these methods become intractable in high-dimensional state spaces due to the large number of samples needed to characterize the distributions. While these approaches (and many others) have had some success, no classic techniques explicitly incorporate the sparsity structure that has been so powerful in modern signal processing.

With the potential to improve signal estimation in many important applications, recent work in compressive sensing has begun to address recovery of time-varying sparse signals. This work can be broadly split into two categories: batch processing and streaming algorithms. Batch processing approaches use all measurements to jointly estimate the states over all time (e.g., [106–110]). More relevant here, however, are algorithms that also seek a way to causally estimate signals (i.e., estimating the current state sequentially as new measurements become available). Within the causal estimation literature, the proposed algorithms can be further divided into algorithms that build off of the traditional Kalman filter equations, algorithms based on modifying the BPDN cost function to have time-dependent terms, and algorithms that include temporal information by modifying the weights in a weighted BPDN optimization.

In the first of these classes, one approach attempts to leverage the Kalman estimator directly by using a pseudo-norm in the update equations to encourage sparser solutions [111], then enforcing an ℓ_1 constraint on the state after the Kalman update. Another

method [112, 113] takes a two-step approach: first performing a support estimation using ℓ_1 cost functions and then running the traditional Kalman equations on a restricted support set. Both approaches essentially modify the Kalman filter equations directly (including the propagation of covariance matrices), despite the statistics of the problem being highly non-Gaussian. From a computational perspective, storing, and inverting full covariance matrices is also prohibitive for the high-dimensional signal problems, where sparsity models have been most successful. Additionally, while the work in [112,113] assume sparse innovations (with the condition that the innovations sparsity is much less than even the state sparsity), the robustness to model mismatch has not been fully explored in these approaches.

More recent approaches (e.g., [15,114–116]) fall into the second category and start from the optimization framework rather than the Kalman update equations, using a restricted dynamic model for the coefficients' temporal evolution. In these approaches, additional norms are appended to the BPDN cost function to include the dynamic state prediction in the estimate. Such approaches make explicit strong assumptions on the innovations statistics and are thus not very robust to model mismatch. Additional models have considered more direct coefficient transition modeling via Markov models [117,118] either by utilizing message passing to propagate support information through time [118] or by using the previous estimate to influence coefficient selection through a modified orthogonal matching pursuit (OMP) [117]. These approaches either incorporate restrictive models designed for specific applications [114–116, 118] (i.e., the approach as specified and implemented restricts the dynamics function to $f(\boldsymbol{x}) = \boldsymbol{x}$), have limited robustness due to the fact that they strictly enforce a support set estimate [15,117] or retain the covariance propagations from the Kalman setting [116]. None of these approaches strike a balance between utilizing dynamic models, adapting to improve robustness to model error, utilizing higher-order statistics native to sparse signal estimation, and retaining the computational efficiency found in either Kalman filtering and optimized BPDN solvers.

The third of these categories has been the least developed of the three, while showing

the most promise. A small number of weighted BPDN estimation schemes that can include prior information has been recently proposed in the literature, with the main approach being to bias support set estimation using constant multiplicative factors. For example, the work in [119] shows the benefits of a binary weighting scheme in BPDN (not RWL1) where on-support and off-support coefficients are weighted by low and high weights, respectively. As an extreme case, the updating scheme in [115] does not penalize on-support values at all but continues to penalize off-support coefficients in the typical BPDN fashion. Other work in [120, 121] uses a reweighted-ℓ_2 scheme with weights scaled by a small constant value if they are expected to be on the support. While these algorithms incorporate general "prior information," they have not been proposed or demonstrated for dynamic filtering.

4.1 Basis Pursuit De-Noising with Dynamic Filtering

4.1.1 Optimization Framework for State Estimation

The framework we present here is based on the formulation of the traditional Kalman filter as a one step optimization problem, i.e only estimates of parameters from the previous iteration can be used in the cost function. In the Kalman filter, the global solution of the state estimation problem for the system described by Equations (35) and (36) is given by the total optimization over the entire time-line

$$\{\widehat{\boldsymbol{x}}_k\}_{k=0}^{n} = \arg\min_{\{\boldsymbol{x}_k\}_{k=0}^{n}} \left[\sum_{k=0}^{n} \|\boldsymbol{y}_k - \boldsymbol{\Phi}_k \boldsymbol{x}_k\|_{\boldsymbol{Q}_k^{-1},2}^2 + \sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{F}_k \boldsymbol{x}_{k-1}\|_{\boldsymbol{R}_k^{-1},2}^2 \right], \tag{37}$$

where $\|\boldsymbol{x}\|_{\boldsymbol{Q},2}^2 = \boldsymbol{x}^H \boldsymbol{Q} \boldsymbol{x}$, $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ are the covariance matrices of of the measurement noise and innovations, respectively. This optimization program can be defined via a Bayesian estimator and results in the standard Kalman filter equations, as demonstrated in Appendix 8.1. The Kalman filter allows us to calculate the latest state estimate $\hat{\boldsymbol{x}}_n$ from the optimization (37) locally using only the previous estimate $\hat{\boldsymbol{x}}_{n-1}$ and its covariance. The optimization program that estimates $\boldsymbol{x}_n$ alone can be written as

$$\widehat{\boldsymbol{x}}_n = \arg\min_{\boldsymbol{x}_n} \left[\|\boldsymbol{y}_n - \boldsymbol{\Phi}_n \boldsymbol{x}_n\|_{\boldsymbol{Q}_n^{-1},2}^2 + \|\boldsymbol{x}_n - \boldsymbol{F}_n \hat{\boldsymbol{x}}_{n-1}\|_{\boldsymbol{P}_{n|n-1}^{-1},2}^2 \right], \tag{38}$$

where $\boldsymbol{P}_{n|n-1}$ is the estimated covariance matrix for time n. Both $\hat{\boldsymbol{x}}_{n-1}$ and $\boldsymbol{P}_{n|n-1}$ are parameters that are calculable iteration-to-iteration. By showing that the solution at iteration n is the same for (37) and (38), the dimension of the optimization to be solved at each iteration is reduced significantly; The dimension of the solution is decreased from nN to N. Additionally, by writing the estimation as an optimization program, we can begin to consider leveraging sparsity by applying appropriate ℓ_1 norms in the same way that ℓ_1 norms are introduced in static least-square cases. One encouraging application in [122] addresses a case where this formulation allows for the mitigation of sparse noise in the measurement equation. We extend this idea to directly incorporate knowledge of sparsity in the innovations and states themselves in the estimation problem.

4.1.2 Sparsity in the Dynamics

In previous work, the assumptions of sparsity in the system has varied. While many have assumed some measure of sparsity in the state itself [106, 112, 118], some have assumed knowledge of sparsity in the innovations [118] as well. Our work here takes both possibilities (sparsity in the state and innovations) and uses the framework presented in order to determine the potential gains that be realized in the context of state estimation by incorporating appropriate ℓ_1 norms. We primarily focus on sparsity in the state evolution equation due to its relevance to specific applications, such as tracking and video. The three models we present are sparse states, sparse innovations and both sparse states and innovations. In adding the regularization terms for each case, we note that only the first order statistic of the previous estimation (the expectation) is taken into account and therefore our optimization programs are not assured to be globally optimal. This differentiates our work from (38) in that the Kalman filter which propagates second order statistics (the covariance matrix of the estimate $\boldsymbol{P}_{n|n-1}$) to obtain a globally optimal solution. As [122] points out, when deviating from the optimization problem (38), this matrix of parameters stops having an interpretation as a covariance matrix. Therefore we do not attempt to estimate second order parameters, and instead only utilize the state estimate.

Sparse States

The first type of sparsity we consider is sparsity in the states only. This model still assumes that our estimate is accurate to a Gaussian random variable (e.g. $\boldsymbol{\nu}_n \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}_N)$), indicating that the predicted dynamics, $f_n(\cdot)$, return a dense estimate. Such a model could potentially be considered, for instance, in a tracking problem where the number of objects to be tracked are relatively small [123]. In this case, we can add an ℓ_1 norm over the state's coefficients $\boldsymbol{z}$ to the update equation (representing our knowledge of the sparsity of the signal). This addition results in the basis-pursuit de-noising with dynamic filtering (BPDN-DF) with an ℓ_2 norm over the innovations,

$$\widehat{\boldsymbol{a}}_n = \arg\min_{\boldsymbol{a}} \left[\|\boldsymbol{y}_n - \boldsymbol{\Phi}_n \boldsymbol{\Psi} \boldsymbol{a}\|_2^2 + \lambda \|\boldsymbol{a}\|_1 + \kappa \|\boldsymbol{\Psi} \boldsymbol{a} - f_n(\boldsymbol{\Psi} \widehat{\boldsymbol{a}}_{n-1})\|_2^2 \right], \tag{39}$$

where λ is the sparsity parameter and κ represents the ratio of the measurement variance to the innovations variance. It is important to note here that while the program (39) does not rely on linear dynamics and performs well in tracking simulations, it is has no assurance for global optimality. Thus for linear dynamics ($f_n(\boldsymbol{x}) = \boldsymbol{F}_n \boldsymbol{x}_n$) Kalman filtering still has assured optimal performance in the steady state tracking regardless of signal sparsity. This is due to the fact that the Kalman in essence is piecewise updating the solution to a larger matrix inverse problem. Given enough measurements, this matrix will be full rank, resulting in a fully determined system. Thus while our program has no assurance of obtaining a better steady-state MSE, we do expect that it will converge faster (when the Kalman filter is still underdetermined).

Sparse Innovations

While including the idea of sparseness in the state is useful during convergence, there is no apparent gain in the steady state MSE over traditional Kalman filters. Where more significant gains over the Kalman filter should be realized is in the case of sparse innovations. The Gaussian assumption is key to the derivation of the Kalman filtering equations, without which the estimate covariance matrix is not exactly and analytically calculable (making the

estimate suboptimal). The sparse innovations model leads to using the ℓ_1 norm on the error of the prediction,

$$\widehat{\boldsymbol{x}}_n = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y}_n - \boldsymbol{\Phi}_n \boldsymbol{x}\|_2^2 + \kappa \|\boldsymbol{x} - f_n(\boldsymbol{x}_{n-1})\|_1, \tag{40}$$

where κ represents the trade off between reconstruction and sparsity. A setup of this type was initially presented in [14], only with a buffer that estimated the past P states at once, effectively smoothing to an extent. In keeping with the fast-update philosophy of Kalman filtering, a homotopy algorithm was used to update states given new measurements, thereby decreasing the time for the update. What is interesting in the optimization program (40) is that under a change of variables $\boldsymbol{\nu}_n = \boldsymbol{x} - f_n(\boldsymbol{x}_{n-1})$ and given a known sparsity on the innovations, the innovations is then recoverable with CS guarantees, given the typical constraints on $\boldsymbol{\Phi}_n$. Thus with perfect knowledge of the previous state, the new state is recoverable with the same guarantees. What is not assured is the convergence of this algorithm from an erroneous initialization to a steady-state estimation error, as would be desired from a tracking algorithm. We show from simulation that it takes more measurements to have (40) converge than either of the algorithms that utilize the state sparsity directly. While obtaining a lower error vs. per-iteration measurement number, [14] shows that when estimating the past P states together, the this model permits a fast update (faster than using BPDN directly) using homotopy steps.

Sparse States and Sparse Innovations

The final case we consider in this paper is the case where both the state and the innovations are sparse. This combination is of the most interest to us due to its application to video where each image can be thought of as sparse in some basis and 'new' objects not predictable from older frames can be thought of as sparse innovations. In this case there are two forms of sparsity that can be leveraged. We can modify (40) to include the sparsity inducing term included in (39),

$$\widehat{\boldsymbol{a}}_n = \arg\min_{\boldsymbol{a}} \left[\|\boldsymbol{y}_n - \boldsymbol{\Phi}_n \boldsymbol{\Psi} \boldsymbol{a}\|_2^2 + \lambda \|\boldsymbol{a}\|_1 + \kappa \|\boldsymbol{\Psi}\boldsymbol{a} - f_n(\boldsymbol{\Psi}\widehat{\boldsymbol{a}}_{n-1})\|_1 \right], \tag{41}$$

where once again λ trades off for sparsity in the state and κ trades off for sparseness in the innovations.

4.1.3 Simulations

We test the optimization programs on randomly generated sequences of temporally evolving signals that include sparsity in the signals and the prediction errors. First, we use a standard Gaussian innovation and compare the standard Kalman filter with the optimizations (39), (40), (41), and BPDN performed independently at each iteration (optimization (3), denoted CS in the figures) to demonstrate the utility of leveraging only the sparsity of the signal. We simulate a 20-sparse state of length 500 evolving by a permutation matrix followed by a scaling matrix (both different at each iteration, and assumed known *a priori*) with zero-mean, 0.001 variance Gaussian innovations. A Gaussian random matrix (different at each step) is used to take 30 measurements at each iteration with i.i.d. zero mean, variance 0.01 measurement noise. For each optimization, λ and κ were chosen by performing a parameter sweep and choosing the best value. For Figure 13 and all subsequent simulations we initialize the state to the zero vector and obtain the expected behavior by averaging over 40 trials.

Figure 13 demonstrates that while the Kalman filter does indeed reach the noise floor after enough iteration, (39) does, as predicted, reach a lower relative MSE (rMSE) during the time frame where Kalman has not yet accumulated enough measurements. Due to the global suboptimality of (39) it does not reach lower steady-state rMSE. However, the tracking error is comparable to that of the Kalman filter which is an optimal solution in this case. What is interesting to note is that (41), the program that attempts to enforce sparsity in the state and the innovations, seems to outperform in both regimes: It obtains a lower steady-state rMSE in less iterations.

To show the performance with sparse innovations, we again estimate a simulated 20-sparse, 500-dimensional vector evolving with the same dynamics as used for Figure 13 with each of the optimization programs presented and compare to independent BPDN, and the

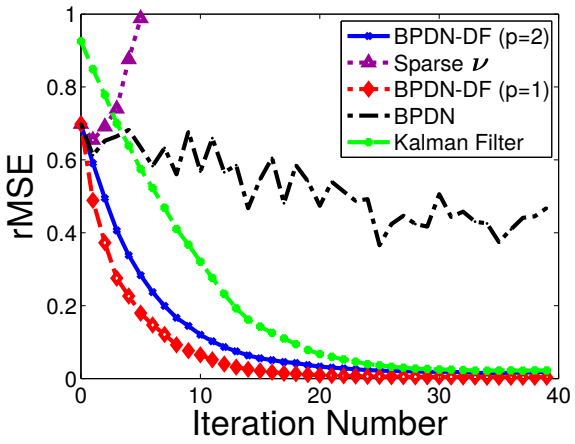

Figure 13: By incorporating the state sparsity in the optimization program, the rMSE converges to its steady-state value faster than a traditional Kalman filter. As expected, independent BPDN performs identically at each iteration and the least matched model (sparsity in the innovations only) diverges in terms of the steady-state rMSE.

Kalman filter. In this case, sparse innovations are introduced via a Poisson random variable with mean 2 (10% of the total number of active coefficients) choosing how many coefficients (chosen at random with a uniform probability over the support) will be switched. This effectively simulates a sparse change in the support of the signal. We allow the system to run for 50 iterations, and record the steady-state rMSE for a different number of random Gaussian measurements. Figure 14 shows that the number of measurements needed (e.g. rows of $\mathbf{\Phi}_n$) for a given steady state tracking error when utilizing both knowledge of sparsity in the state and innovations is significantly less than using any other method. For this program, 60 measurements is sufficient to obtain an rMSE of approximately 3%, while with the same number of measurements independent CS has approximately 17% rMSE and both models which assume Gaussian innovations have much higher steady-state rMSE values.

Figure 15 shows results using an identical setup to Figure 14, only fixing the number of measurements at $M = 80$ and sweeping the mean number of coefficients changed (half the effective sparsity of $\boldsymbol{\nu}$). We see that the optimization in (41) again performs better in terms of the steady-state rMSE. Independent CS recovery performs as expected (the rMSE

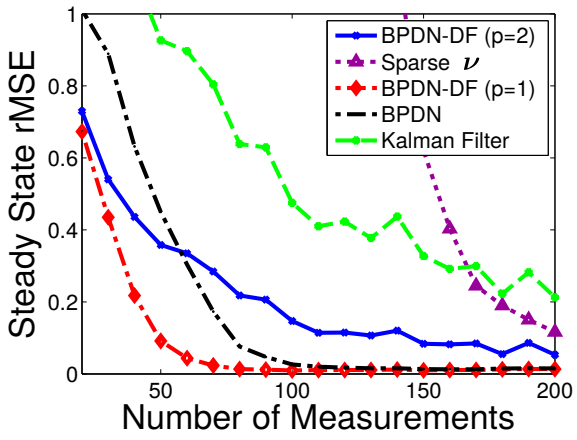

Figure 14: Without Gaussian noise, the Kalman filter has significant trouble tracking the evolving signal, and requires more measurements than any optimization program which takes the sparsity of the signal into account, including independent BPDN. Only using sparse innovations does not outperform any model for small numbers of measurements, but converges quickly for $M > 150$.

is independent of innovations), and both models using Gaussian noise obtain very high errors very quicky with the sparsity of $\boldsymbol{\nu}$. The optimization (40) is not shown here due to its inability to converge to a steady state error with only $M = 80$ measurements per iteration. It would seem that as $\boldsymbol{\nu}$ became more dense, the Gaussian model would be a better fit, but the energy over the support of $\boldsymbol{\nu}$ is on the order of the energy on the support in the state itself, so the sparsity knowledge is required to tease the two apart.

Finally, we can test the situation where only the innovations are sparse, and the state is dense. Such situations can arise in traditional tracking situations, where the number of targets is known, however the deviation in the acceleration, position, or velocity of the targets may suddenly change drastically from an established dynamical model. We similarly generate sequences of signals as used to generate the data in Figures 14 and 15 ($N = 500$) but we set the "sparsity" to $S = 500$, allowing the states to be dense. We then let the innovations at each time-step be 20-sparse, inducing a "shot-noise" very different from Gaussian innovations. At each time-step we take $M = 200$ random Gaussian measurements. Figure 16 depicts the average convergence of BPDN-DF for $p = 1$ and $p = 2$, as well as the

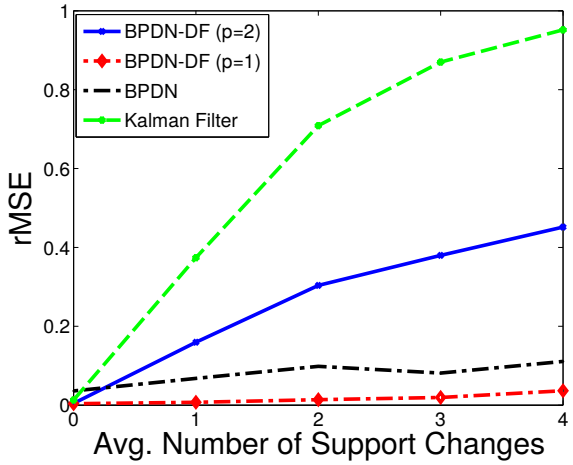

Figure 15: The optimization taking both sparsity in the state and innovations retained the lowest steady-state rMSE for more increased innovations sparsity given a fixed number of measurements ($M = 80$). The performance for BPDN remains constant, as expected, and the performance for the models dependent on Gaussian innovations degrades quickly with additional support deviations from the expectation.

Kalman Filter and time-independent BPDN behavior on these test sequences. The sparse innovations optimization achieves the lowest steady-state rMSE, as it matches the signal statistics exactly. Additionally, the sparse-innovations only optimization appears to be the only tracking method that shows any real convergence behavior at all.

4.2 Guarantees on Basis Pursuit De-Noising with Dynamic Filtering

In basis-pursuit de-noising dynamic filtering (BPDN-DF) [15, 116] we seek an efficient method that can solve a modified Kalman filtering optimization with an added sparsity regularizer. While in general the dynamics inducing norm in Equation (41) can be any p-norm,

$$\widehat{\boldsymbol{a}}_n = \arg\min_{\boldsymbol{a}} \left[||\boldsymbol{y}_n - \boldsymbol{\Phi}_n \boldsymbol{\Psi} \boldsymbol{a}||_2^2 + \lambda ||\boldsymbol{a}||_1 + \kappa ||\boldsymbol{\Psi}\boldsymbol{a} - f_n(\boldsymbol{\Psi}\widehat{\boldsymbol{a}}_{n-1})||_p^p \right],$$

here we consider an ℓ_2-norm penalized innovations term. Guarantees for $p \neq 2$ would be useful (i.e. for $p = 1$ in Section 4.1.2 for sparse innovations), however would require new, different analysis tools. As depicted in Figure 12(c), BPDN-DF essentially balances the prediction via the dynamics model with the measurements and the sparsity assumption to

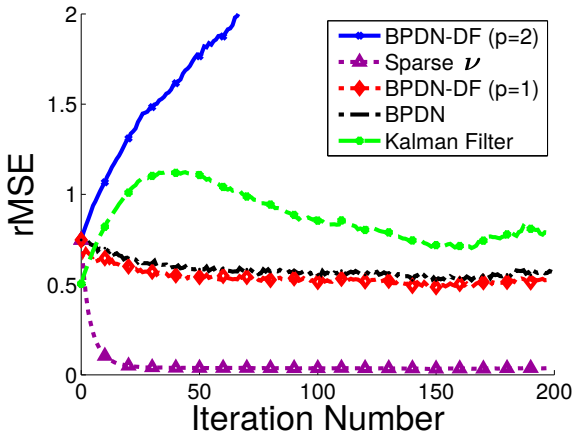

Figure 16: The overall performance with BPDN-style dynamic filtering depends heavily on the state and innovations statistics. While when the innovations is Gaussian, algorithms that leverage state statistics can obtain faster convergence to steady-state and slightly better steady-state rMSE, when the innovations is sparse, the algorithms that assume Gaussian statistics perform poorly. In fact, when presented with dense states with sparse innovations, only the optimization that explicitly utilizes those statistics obtains a good steady-state rMSE. Even the optimization program that utilized sparse innovations, but also tries to force sparse state statistics fails to recover the time-varying signal with any fidelity.

obtain an estimate of the current state.

4.2.1 General Convergence Guarantees

Our first result is summarized in the following theorem:

Theorem 5. *Suppose that at each time-step n, $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ satisfies RIP(2K,δ), $\gamma > 0$ and $\kappa > 0$ are known constants. Additionally, suppose that the dynamics function $f(\cdot)$ satisfies $\|f(\boldsymbol{a}_1) - f(\boldsymbol{a}_2)\|_2 \leq f^* \|\boldsymbol{a}_1 - \boldsymbol{a}_2\|_2$ and that for all $n \geq 0$ the error and innovations satisfy $\|\boldsymbol{\epsilon}_n\|_2 \leq \bar{\epsilon}$ and $\|\boldsymbol{\nu}_n\|_2 \leq \bar{\nu}$. Under these conditions, the result of solving the optimization program of Equation* (41) *satisfies*

$$\|\widehat{\boldsymbol{a}}_n - \boldsymbol{a}_n\|_2 \leq \beta^n \left(\|\boldsymbol{e}_0\|_2 - \frac{\alpha}{1-\beta} \right) + \frac{\alpha}{1-\beta},$$

a where the constant α is given by

$$\alpha = C_1 \frac{1}{\sqrt{1+\kappa}} \bar{\epsilon} + C_1 \sqrt{\frac{\kappa}{1+\kappa}} \bar{\nu} + C_2 \frac{\gamma}{1+\kappa} \sqrt{q},$$

and the linear convergence rate is

$$\beta = C_1 \sqrt{\frac{\kappa}{1+\kappa}} f^*,$$

and the constants C_1 and C_2 are the constants from the bounds on solving the static BPDN problem with sparsity K and a modified RIP parameter $\widetilde{\delta} = \boldsymbol{\delta}/(1+\kappa)$.

The proof for Theorem 6 is provided in Appendix 8.2. This theorem essentially states that BPDN-DF is guaranteed to converge at a linear rate β so long as $\beta < 1$. Solving for κ in this constraint gives us an upper bound on κ

$$\kappa < \frac{1}{(C_1 * f^*)^2 - 1} \qquad C_1 * f^* > 1$$
$$\kappa > \frac{1}{1 - (C_1 * f^*)^2} \qquad C_1 * f^* < 1$$

which guarantees that there will be a range of parameters for which the algorithm is stable. In the first condition, a larger f^* requires a smaller C_1 value to have the same rage of admissible κ values. This means that less smooth dynamics functions need a more accurate BPDN solver to stay stable. Likewise, a less accurate solver requires a smoother dynamics function to be stable for the same κ range. In the second of these conditions, κ must be greater than a negative number, which implies that all positive κ values result in a stable algorithm.

We validate our bound by comparing to the empirical behavior of BPDN-DF. We run BPDN-DF on sequences of 100 $K = 15$-sparse signals of size $N = 576$. At each time step we take $M = 68$ measurements. We recover the sequence of signals using BPDN-DF with $\gamma = 5.5\times10^{-4}$ and sweep κ over 30 possible values. We average all our results over 50 trials. We fit our theoretical bounds by selecting C_1 and C_2 such that they fall above the empirical curves. Figure 17 shows That the convergence time increases as predicted by the theory ($n_{\text{convergence}} \propto \log^{-1}\beta$). The worst-case-scenario nature of the bound, however, creates a gap in the predicted steady-state error. The theoretical curve for the error does not predict the dip that occurs for the optimal κ value, and instead has a monotonically increasing

value from $\kappa = 0$, the point that corresponds to simply running BPDN independently at each iteration.

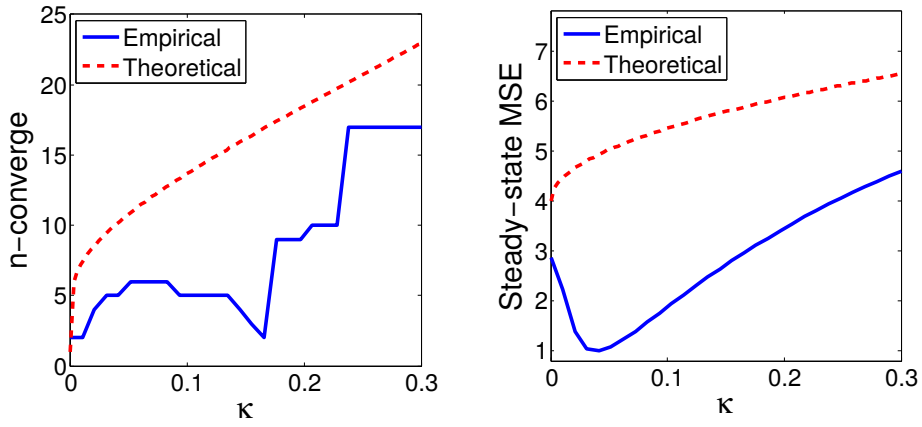

Figure 17: The theoretical bound was fit to empirical curves of BPDN-DF's behavior as a function of κ. Top: The empirical number of iterations to convergence (solid blue curve) generally increases as a function of κ, as predicted by theory (dashed red curve). The dip in the empirical curve corresponds to the crossover point as the steady-state error increases from being below the initial error to being above the initial error. Bottom: The derived bound accounts for the worst possible recovery at each time-step, and thus yields an extreme upper bound in terms of the steady-state error.

4.2.2 ISTA based convergence

One detriment of Theorem 6 is that the innovations and measurement error both effect the overall error bound through constants involving the RIP conditioning δ. The information obtained through the dynamics function, however, does not involve the measurement matrix Φ, and therefore the innovations is not distorted by an RIP-matrix. This means that the innovations terms should not involve δ and Theorem 6 can be strengthened.

While general guarantees would treat the whole concatenated ℓ_2 norm as a full matrix that has a joint RIP constant, we can instead use a method for determining the optimization program's accuracy that analyzes a specific optimization procedure. Specifically, in

keeping with the philosophy of fast, efficient optimization, we analyze the iterative soft-thresholding method described in [124, 125]. For BPDN-DF, the ISTA optimization program can be written as the following iterative procedure over the algorithmic time l,

$$\begin{aligned} \boldsymbol{u}_n^{l+1} &= \widehat{\boldsymbol{a}}_n^l + \frac{\eta}{1+\kappa}\boldsymbol{\Psi}^T\left(\boldsymbol{\Phi}_n^T\left(\boldsymbol{y}_n - \boldsymbol{\Phi}_n\boldsymbol{\Psi}\widehat{\boldsymbol{a}}_n^l\right) + \kappa\left(f(\boldsymbol{\Psi}\widehat{\boldsymbol{a}}_{n-1}) - \boldsymbol{\Psi}\widehat{\boldsymbol{a}}_n^l\right)\right) \\ \widehat{\boldsymbol{a}}_n^{l+1} &= T_\gamma(\boldsymbol{u}_n^{l+1}), \end{aligned} \tag{42}$$

where, as in Equation 69, $\boldsymbol{u}$ is the un-thresholded version of the signal which gets updated by the error residual at each algorithmic time-step l, and η is the algorithm's step size.

To determine convergence and accuracy guarantees on this algorithm and cost function, we leverage recent techniques employed in [125], which have shown accuracy and convergence guarantees on ISTA when solving BPDN with no dynamics term. We modify this previous work to account for the fact that in BPDN-DF, as opposed to plain BPDN, only part of the ℓ_2 portion of the optimization is affected by the RIP of the measurements. The other portion only depends on the properties of the dynamics function $f(\cdot)$. We can obtain in this way a bound both for the convergence and steady-state error of ISTA applied to BPDN-DF, as summarized in the following theorem,

Theorem 6. *Suppose that at each time iteration n, $\boldsymbol{\Phi}_n$ satisfies RIP(2S,δ), the dynamics satisfies $\|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\|_2 \leq f^* \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2$, the error and innovations satisfy $\|\boldsymbol{\epsilon}_n\|_2 \leq \overline{\epsilon}$ and $\|\boldsymbol{\nu}_n\|_2 \leq \overline{\nu}$, and the coefficient energy is bounded by b. If $\gamma > 0$, $\kappa > 0$ are known constants, $\eta < 2/(1+\delta)$ is the ISTA step size and the following condition is met:*

$$\kappa\left((1 - |\eta - 1| - \eta f^*)b - \gamma\sqrt{q} - \eta\overline{\nu}\right) \geq \gamma\sqrt{q} + \eta\sqrt{1+\delta}\overline{\epsilon} - (1 - |\eta - 1| - \eta\delta)\, b,$$

Then the solution to Equation (41) *obtained via ISTA satisfies*

$$\left\|\widehat{\boldsymbol{a}}_n - \boldsymbol{a}_n\right\|_2 \leq \beta^n\left(\|\boldsymbol{e}_0\|_2 - \frac{\alpha}{1-\beta}\right) + \frac{\alpha}{1-\beta},$$

where the steady-state error is given by

$$\frac{\alpha}{1-\beta} = \frac{(1+\kappa)\gamma\sqrt{q} + \eta\sqrt{1+\delta}\overline{\epsilon} + \eta\kappa\overline{\nu}}{(1+\kappa)(1 - |\eta - 1|) - \eta\delta - \eta\kappa f^*},$$

where q is a constant that depends on the sparsity S[2], and the linear convergence rate is

$$\beta = \frac{\eta\kappa f^*}{(1 - |\eta - 1|)(1 + \kappa) - \eta\delta},$$

The proof of Theorem 6 is outlined in Appendix 8.3, and essentially deduces a difference equation for the estimate accuracy at each algorithmic and temporal time-step, which can be solved for the error bound at any time. The first thing we note about Theorem 6 is that if we set $\kappa = 0$, we obtain exactly the results in [125] for solving BPDN with no dynamic filtering term. Next we can see that the resulting convergence rate implies that ISTA only converges for the BPDN-DF cost function when $\beta < 1$. Since most parameters are system or signal dependent, and are not controllable, we can interpret this requirement as a condition on the cost function parameters γ and κ. In particular, since γ does not appear in the expression for β, we can consider this to be a bound on κ,

$$\kappa < \frac{1 - |1 - \eta| - \eta\delta}{\eta f^* + |\eta - 1| - 1} \quad \text{if} \quad \eta f^* > 1 - |\eta - 1|, \tag{43}$$

This condition essentially compares the smoothness of the dynamics function with the RIP conditioning of the measurements. For example, as $\eta\delta$ becomes closer to $1 - |\eta - 1|$, the allowable range of κ is pushed towards smaller values, indicating that the dynamics should be emphasized less in BPDN-DF. Alternatively, as ηf^* becomes closer to $1 - |\eta - 1|$, the range of κ becomes pushed towards larger values, indicating that the dynamics can be emphasized more in the optimization cost. Interestingly if $\eta f^* \leq 1 - |\eta - 1|, \beta < 1$ incurs no additional restrictions on κ, as the condition that $\eta < 2/(1 + \delta)$ ensures that the numerator of Equation (43) is positive, and with a negative denominator, this condition simply states that κ must be greater than a negative number. Since κ must already be positive, this condition is redundant.

In terms of the steady-state error, we can see that the bound depends on both the maximum measurement error energy $\overline{\epsilon}$ and the maximum innovations energy $\overline{\nu}$. The parameter

[2]The exact relationship between q and S is actually quite involved, and more details can be found in [125].

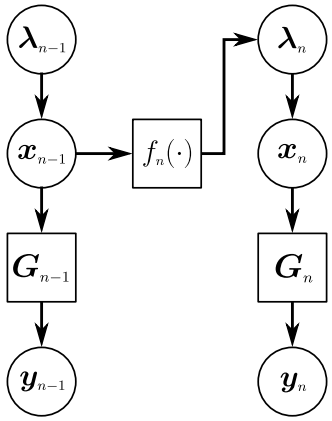

Figure 18: The RWL1-DF algorithm inserts the dynamic information at the second layer of the LSM model for each time step. The graphical model depicts the model dependencies, where prior state estimates are used to set the hyperpriors for the second level variables controlling the variances (i.e., SNRs) of the state estimates at the next time step.

κ trades off between the two, where the trade-off takes into account the RIP conditioning δ as well as the dynamics smoothness f^*.

4.3 Re-Weighted ℓ_1 Dynamic Filtering

Model and Algorithm

In this section we describe the proposed RWL1-DF algorithm for the general dynamics model in (35) with the linear measurement process in (36). The main idea of the proposed method is to use the rich signal description available in a hierarchical sparsity model to propagate second-order uncertainty in dynamic signal estimation (akin to the covariance matrices in Kalman filtering). This approach leverages the LSM model and its connections to RWL1 optimization to build a computationally efficient causal estimator. The main technical innovation we propose is to use the hyper-priors in the LSM model to inject dynamic information into the sparsity inducing priors, using the variable coefficient SNRs to encourage or discourage (but not force) coefficient activity based on predictions from the previous state estimate. The resulting estimation procedure is depicted in Figure 12(d) and the graphical model is depicted in Figure 18.

Specifically, similar to the original LSM, the proposed model describes the conditional

distribution on the sparse coefficients $\boldsymbol{a}$ at time k as a zero-mean Laplacian with different variances:

$$\boldsymbol{a}_k[i] \mid \boldsymbol{\lambda}_k \sim \frac{\lambda_0 \boldsymbol{\lambda}_k[i]}{2} e^{-\lambda_0 \boldsymbol{\lambda}_k[i] |\boldsymbol{a}_k[i]|}. \tag{44}$$

We also set the scale variables controlling the coefficient variance to be Gamma distributed:

$$\boldsymbol{\lambda}_k[i] \sim \frac{\boldsymbol{\lambda}_k^{\alpha-1}[i]}{\boldsymbol{\theta}_k^{\alpha}[i]\Gamma(\alpha)} e^{-\boldsymbol{\lambda}_k[i]/\boldsymbol{\theta}_k[i]}, \tag{45}$$

and allow each of these variables to have different means, ($\alpha\boldsymbol{\theta}_k[i] = E[\boldsymbol{\lambda}_k[i]]$) by modifying the value of $\boldsymbol{\theta}_k[i]$.

The expected value of each scale variable is set based on dynamic information from the previous state estimate. In particular, if the model prediction of $\boldsymbol{a}_k[i]$ based on the previous state is large (or small), the variance of that coefficient at the current estimate is made large (or small) by making $\boldsymbol{\lambda}_k[i]$ small (or large). Large variances allow the model flexibility to choose from a wide-range of non-zero values for coefficients that are likely to be active and small variances (with a mean of zero) encourage the model to drive coefficients to zero if they are likely to be inactive. However, by "encouraging" the model through the use of second order statistics (instead of forcing the model to use a particular subset of coefficients through a separate support estimation process) the model remains robust and flexible. In this work we specifically choose

$$\boldsymbol{\theta}_k[i] = \xi \left(|\boldsymbol{\Psi}^{-1} f_k\left(\boldsymbol{\Psi}\widehat{\boldsymbol{a}}_{k-1}\right)[i]| + \eta \right)^{-1},$$

where $\boldsymbol{\Psi}^{-1} f_k\left(\boldsymbol{\Psi}\widehat{\boldsymbol{a}}_{k-1}\right)[i]$ is the i^{th} coefficient of the previous signal propagated through the dynamics, η is a linear offset and ξ is a multiplicative constant. Note that any general model for the dynamics is allowable and we are not restricting the system to linear dynamical systems. Note also that the absolute values are necessary because $\boldsymbol{\lambda}$ determines a variance, which must be strictly positive. The parameter η determines the distribution of the variance when the coefficient is predicted to be zero, resulting in $\boldsymbol{\Psi}^{-1} f_k\left(\boldsymbol{\Psi}\widehat{\boldsymbol{a}}_{k-1}\right)[i] = 0$. This parameter reflects the magnitude of the innovations in the erroneous model predictions. The joint

MAP estimate of all model parameters becomes

$$[\widehat{\boldsymbol{a}}_k, \widehat{\boldsymbol{\lambda}}_k] = \arg\min_{[\boldsymbol{a},\boldsymbol{\lambda}]} ||\boldsymbol{y}_k - \boldsymbol{\Phi}_k\boldsymbol{\Psi}\boldsymbol{a}||_2^2 + \beta \sum_i |\boldsymbol{\lambda}[i]\boldsymbol{a}[i]|$$
$$-\alpha \sum_i \log(\boldsymbol{\lambda}[i]) + \frac{\lambda_0}{\xi} \sum_i \boldsymbol{\lambda}[i] \left(|\boldsymbol{\Psi}^{-1} f_k(\boldsymbol{\Psi}\widehat{\boldsymbol{a}}_{k-1})[i]| + \eta \right), \tag{46}$$

which is identical to the MAP estimate in the LSM except for the appearance of α and the time-dependencies on the parameters.

As in the LSM case, the optimization in (46) is not easily solved for both $\boldsymbol{a}_k$ and $\boldsymbol{\lambda}_k$ jointly, but the model yields a simple form when using an EM approach. The precise steps of the iteration in this case are

$$\text{E step:} \qquad \widehat{\boldsymbol{\lambda}}_k^t = E_{p(\boldsymbol{\lambda}|\widehat{\boldsymbol{a}}_k^t)}[\boldsymbol{\lambda}] \tag{47}$$

$$\text{M step:} \qquad \widehat{\boldsymbol{a}}_k^t = \arg\min_{\boldsymbol{a}_k} -\log\left[p\left(\boldsymbol{a}_k | \widehat{\boldsymbol{\lambda}}\right)\right] \tag{48}$$

where t denotes the EM iteration number and $E_{p(\boldsymbol{\lambda}|\widehat{\boldsymbol{a}}_k)}[\cdot]$ denotes the expectation with respect to the conditional distribution $p(\boldsymbol{\lambda}|\widehat{\boldsymbol{a}}_k)$. We can write the maximization step as

$$\widehat{\boldsymbol{a}}_k^t = \arg\min_{\boldsymbol{a}} ||\boldsymbol{y}_k - \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{a}||_2^2 + 2\sigma_\epsilon^2\lambda_0 \sum_i \left| \widehat{\boldsymbol{\lambda}}^t[i]\boldsymbol{a}[i] \right|, \tag{49}$$

since the MAP optimization conditioned on the λ parameters reduces to a weighted ℓ_1 optimization.

While the expectation step is often difficult to calculate, this model admits a simple closed-form solution. First we can use the conjugacy of the Gamma and Laplacian distributions to calculate the conditional distribution

$$p(\boldsymbol{\lambda}|\boldsymbol{a}_k) = \frac{p(\boldsymbol{a}_k|\boldsymbol{\lambda})\, p(\boldsymbol{\lambda})}{p(\boldsymbol{a}_k)},$$

which is separable in $\boldsymbol{\lambda}$. We can analytically write this distribution by evaluating

$$\begin{aligned} p(\boldsymbol{a}_k[i]) &= \frac{\alpha\theta\lambda_0}{2(\theta\lambda_0|\boldsymbol{a}_k[i]| + 1)^{\alpha+1}} \\ &= \frac{\alpha\left(\xi^{-1}|\boldsymbol{\Psi}^{-1} f_k(\boldsymbol{\Psi}\widehat{\boldsymbol{a}}_{k-1})[i]| + \xi^{-1}\eta\right)^{-\alpha}}{2(\lambda_0|\boldsymbol{a}_k[i]| + \xi^{-1}|\boldsymbol{\Psi}^{-1} f_k(\boldsymbol{\Psi}\widehat{\boldsymbol{a}}_{k-1})[i]| + \xi^{-1}\eta)^{\alpha+1}}, \end{aligned} \tag{50}$$

and the expectation can be calculated as

$$\begin{aligned} E_{p(\boldsymbol{\lambda}|\boldsymbol{a}_k)}[\boldsymbol{\lambda}] &= \frac{(\alpha+1)\theta}{\theta\lambda_0|\boldsymbol{a}_k[i]|+1} \\ &= \frac{(\alpha+1)\xi}{\xi\lambda_0|\boldsymbol{a}_k[i]|+|\boldsymbol{\Psi}^{-1}f_k(\boldsymbol{\Psi}\widehat{\boldsymbol{a}}_{k-1})[i]|+\eta}. \end{aligned} \tag{51}$$

Putting the pieces of the EM algorithm above together results in an iterative re-weighted ℓ_1 dynamic filter (RWL1-DF):

$$\widehat{\boldsymbol{a}}_k^t = \arg\min_{\boldsymbol{a}}\left[||\boldsymbol{y}_k-\boldsymbol{\Phi}_k\boldsymbol{\Psi}\boldsymbol{a}||_2^2+2\sigma_\epsilon^2\lambda_0\sum_i\left|\boldsymbol{\lambda}_k^t[i]\boldsymbol{a}[i]\right|\right], \tag{52}$$

$$\boldsymbol{\lambda}_k^{t+1}[i] = \frac{2\tau}{\beta|\widehat{\boldsymbol{a}}_k^t[i]|+|\boldsymbol{\Psi}^{-1}f_k(\boldsymbol{\Psi}\widehat{\boldsymbol{a}}_{k-1})[i]|+\eta}, \tag{53}$$

where $\tau = (\alpha+1)\xi$ is a constant scaling value, $\beta = \lambda_0\xi$ can be interpreted as a trade-off between the measurement and the prediction, and the signal of interest can again be recovered via $\widehat{\boldsymbol{x}}_k = \boldsymbol{\Psi}\widehat{\boldsymbol{a}}_k$. The resulting optimization procedure looks nearly identical to the static RWL1 algorithm except that the denominator in the $\boldsymbol{\lambda}$ update contains a term depending on the previous state. This term encourages smaller $\boldsymbol{\lambda}$ values (i.e., higher variances) in the elements that are predicted to be highly active according to $f_k(\widehat{\boldsymbol{x}}_{k-1})$. This graduated encouragement of coefficients selected by the prediction (rather than direct penalization) allows the algorithm to perform especially well when the states and innovations are sparse while retaining good performance when the innovations are denser. Furthermore, the simple form means that the explosion of recent work in ℓ_1 optimization methods can be leveraged for computationally efficient recursive updates. In particular, since no covariance matrix inversion is required and many modern ℓ_1 estimation methods require only matrix multiplication (and no inversion), this approach is also amenable to high-dimensional data analysis.

Convergence and Stability

Despite being highly nonlinear, we can demonstrate some stability and convergence properties of the proposed algorithm. First, the RWL1-DF algorithm is stable in the sense that

the estimates of interest (i.e., the coefficient and variance estimates) are guaranteed to be bounded. This property follows directly from the update equations in (52) and (53). At a given time step n, for each EM iteration t we can immediately see that the variance estimates are within the range $\boldsymbol{\lambda}_k^{t+1}[i] \in (0, 2\tau/\eta]$. With these variances, the weighted BPDN optimization in (53) will yield a solution where the output coefficients are also bounded. In the worst case, if an intermediate coefficient estimate is transiently very large (i.e., tending towards infinity), the variances would tend to zero and the subsequent iteration of (53) would be a standard least-squares estimate (which is finite).

General properties of the EM algorithm can also be used to provide some convergence guarantees for the proposed algorithm. Specifically, existing guarantees for the EM algorithm can be used to show that the EM iterations in proposed algorithm (i.e., an estimate at a single time step) have coefficient differences that asymptotically converge to zero, $\lim_{t\to\infty} \|\boldsymbol{a}_n^t - \boldsymbol{a}_n^{t+1}\| = 0$ [126]. Stronger convergence results (i.e., the convergence of the estimate to a fixed value) require continuous derivative of the objective function, which is not the case here due to the presence of the ℓ_1 norm. While some results pertaining to the convergence and accuracy of the RWL1 algorithm are known [127], these results hold only in the limit as the denominator of the weight updates approaches $|\boldsymbol{a}_n|$ (i.e. $|\boldsymbol{\Psi}^{-1} f_k(\boldsymbol{\Psi}\widehat{\boldsymbol{a}}_{k-1})[i]| + \eta \to 0$) and therefore are not applicable in the present case. Despite the lack of stronger convergence guarantees (which are often difficult to establish in non-smooth problems), the numerical results in section 5.3 demonstrate that the algorithm converges with just a few EM iterations in practice.

4.4 Dynamic Filtering Simulations

While the RWL1-DF inference scheme is a general inference tool with many potential applications, we focus our evaluation purposes on CS recovery. Compressive sensing offers systematic ways to specify inference problems (ratios of M to N with respect to the signal sparsity) that are very challenging for static inference techniques and adequate estimates

require exploiting dynamic information. In the examples below we implement CS recovery in both a stylized tracking scenario with synthetic data and an example on a natural video sequence.

In all simulations, we compare the performance of RWL1-DF to existing algorithms from the literature where possible, noting in each particular case where algorithms were unable to be evaluated because they are incompatible or computationally prohibitive. Standard Kalman filtering is not shown because it performs very poorly in these type of simulations (i.e., it doesn't converge to a stable estimate with the sparse statistics of the applications we use) [15]. The performance of independent BPDN (BPDN applied independently at each time step with no temporal information) and independent RWL1 (RWL1 applied independently at each time step with no temporal information) are also shown to highlight the benefit of including dynamic information. The most illustrative comparison is with BPDN-DF (BPDN modified, as in Equation (41) [15, 116]) due to its similarity in implementation and philosophy to RWL1-DF.

4.4.1 Stylized tracking scenario

To explore the performance and robustness of RWL1-DF in detail, we first perform inference on synthetic data that simulates a stylized tracking scenario. The use of synthetic data provides us with "ground truth" so we can make controlled variations of the data characteristics. In this data, we generate an image with S non-zero moving pixels of various intensities that move with time and represent targets that must be tracked. The movement of these non-zero pixels $\boldsymbol{F}_k$ is specified to be constant motion, and the simulated dynamics includes a sparse innovations term (i.e., dynamic model error) that causes target motion to change in each time step for some percentage of the pixels p. In other words, at every time step there is a probability p of each target abruptly changing directions to violate the dynamics model assumed by the inference algorithms. This process simulates an innovation that is approximately $2Sp$-sparse at every iteration, allowing us to evaluate the algorithm's robustness to a type of model mismatch (i.e., shot noise) that is particularly challenging for

Kalman filter techniques.

We evaluate RWL1-DF by using it to track these moving pixels from M compressive Gaussian measurements. This simple test captures the model notions of sparse state elements that have changing support locations and values with time, with a significant degree of model mismatch (i.e., sparse innovations). With full control over the synthetic data, we can evaluate the performance of recovery algorithms in detail, including temporal convergence properties and variations in model parameters (e.g., number of measurements, degree of model mismatch, noise level, etc.). Note that this scenario is particularly challenging for many existing algorithms because of the arbitrary model dynamics (e.g., $\boldsymbol{F}_k \neq I$) that may vary with time. In particular, comparisons with the algorithms described in [114, 115, 119, 120] (or modifications of them to accommodate arbitrary dynamics) were attempted and are not shown here because they still performed significantly worse than static estimation (e.g., BPDN) even after extensive searching for good parameter settings.

Specifically, we create 24x24 pixel videos ($N = 24^2$) with 20 moving particles ($S = 20$). The vectors are observed with Gaussian measurement matrices (with normalized columns) that are independently drawn at each iteration, and we add Gaussian measurement noise with variance $\sigma_{\boldsymbol{\epsilon}}^2 = 0.001$. We vary the number of measurements to observe the reconstruction capability of the algorithm in highly undersampled regimes, but the number of measurements per time step is always constant within a trial. All simulations average the results of 40 independent runs and display reconstruction results as the relative mean-squared error (rMSE) for each frame, calculated as:

$$\frac{\left\|\boldsymbol{x}_k - \widehat{\boldsymbol{x}}_k\right\|_2^2}{\|\boldsymbol{x}\|_2^2}. \tag{54}$$

For independent BPDN, at each iteration we use the value $\lambda = 0.55\sigma_{\boldsymbol{\epsilon}}^2$. For independent RWL1 we use $\lambda_0 = 0.0011$, $\tau = 1$ and $\nu = 0.01$. For BPDN-DF we use $\gamma = 0.5\sigma_{\boldsymbol{\epsilon}}^2$ and $\kappa = 0.0007/(p+1)$. For RWL1-DF we use $\lambda_0 = 0.0011$, $\tau = 1$ and $\nu = 1 - 2p/S$. These parameters were optimized using a manual parameter sweep. Furthermore, for comparison we

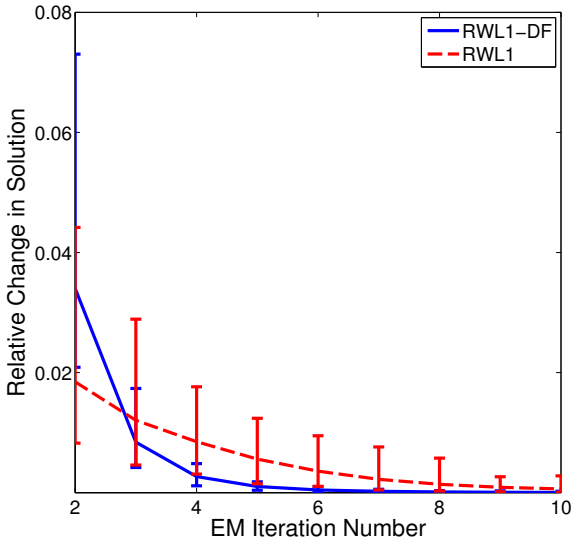

Figure 19: Convergence of RWL1 and RWL1-DF. The mean relative change (over 200 frames) in the coefficients is plotted, with the error bars indicating the maximum and minimum values. The relative norm difference of the coefficients in the RWL1 and RWL1-DF algorithms falls quickly over the first 10 iterations. The dynamic information helps the RWL1-DF algorithm converge faster, requiring approximately 5-7 iterations to converge.

also show the performance of an optimal oracle least-squares solution, where the support at each iteration is known ($\mathbf{\Phi}_k$ becomes overdetermined).

The iterative algorithms based on RWL1 are stopped when the relative norm-squared difference between coefficients at consecutive iterations $\|\boldsymbol{a}_n^t - \boldsymbol{a}_n^{t+1}\|_2^2 / \|\boldsymbol{a}_n^t\|_2^2$ falls below a specified threshold (we use 0.1%). In practice, we see that this EM convergence happens in just a few (typically 5-7) iterations. Figure 19 shows the relative coefficient change over EM iteration, demonstrating that RWL1 (i.e. without dynamics) convergence occurs by 10 iterations and RWL1-DF actually converges faster due to the improved performance from incorporating dynamic information.

Figure 20(a) shows a single trial with $M = 80$ measurements and $2Sp = 5$ innovation errors at each time step. The estimation provided by BPDN-DF and RWL1-DF improves with time and converges to steady state values, indicating that both approaches are exploiting useful dynamic information to sequentially improve over time (in contrast to the methods that do not incorporate dynamic information, as expected). Note that RWL1-DF

reaches substantially lower steady-state recovery error than BPDN-DF, illustrating the net improvements gained by using second-order statistics in the estimation.

To explore the performance of RWL1-DF, Figure 20(b) displays the results of varying the number of measurements while holding $2Sp = 5$. While performance for all algorithms becomes more comparable for large numbers of measurements, it is clear that exploiting temporal information can most improve performance in the highly undersampled regime. In particular, RWL1-DF is able to sustain virtually the same steady-state rMSE down to much more aggressive levels of undersampling than BPDN-DF.

Finally, we explore the robustness of each algorithm to model errors by fixing the number of measurements ($M = 70$) and varying the sparsity of the innovations $2Sp$. Figure 20(c) shows the results, illustrating that RWL1-DF uses the second-order statistics to sustain better performance than BPDN-DF when the innovations are sparse (i.e., shot noise). We note that when $2Sp > 8$, where RWL1-DF results in much higher rMSE errors, the total number of model errors is 50% of the signal sparsity and may be better approximated by a dense (i.e., non-sparse) innovations model.

4.4.2 CS recovery of natural video sequences

To test the utility of RWL1-DF on natural signals, we explore its performance on a simulation of compressively sampled natural video sequences. These results will report in-depth comparison of a single challenging video sequence (the Foreman sequence[3]) as well as aggregate statistics from a batch of video from a BBC nature documentary (as used in [128]). The documentary footage is valuable as broad comparison because it contains many different types of motion, including static frames with localized changes and highly dynamic frames with moving subjects across large portions of the visual field.

In our simulation of CS video recovery, we take the time-varying hidden state $\boldsymbol{x}_n$ to be the wavelet (synthesis) coefficients at each frame of the video. While the true frame-by-frame dynamics of natural video are likely to be complex and non-linear, for this simulation

[3]The Foreman sequence is available at: http://www.hlevkin.com/TestVideo/foreman.yuv

we use a simple first-order model predicting that the coefficients will remain the same from one frame to the next: $f_k(\boldsymbol{x}) = \boldsymbol{x}$ for all k. While this model is very simple and could certainly be improved, the objective of this simulation is to evaluate the inference performance of many algorithms under the same model. An important aspect of this evaluation is the robustness of the algorithms when the dynamic model is incorrect, which will certainly be true for this static model under significant movement in the video sequence.

We note that RWL1-DF is not specialized to this simple dynamics model, and an improved model (which could easily be incorporated into RWL1-DF) would presumably improve the recovery performance of any inference algorithm. A additional benefit to assuming stationary dynamics is that it allows us to compare recovery performance with a number of existing algorithms that do not currently have arbitrary dynamics as part of the approach, including DCS-AMP [129] and modCS [115]. To demonstrate the advantage of using the coefficient values instead of only support information, we also compare to a modified version of the algorithm described in [119]. Specifically, the approach described in [119] weights the coefficients in BPDN with binary values based on an estimate of the support set, and we modify this approach to use a support set prediction based on the model dynamics. We call this approach WL1P for weighted ℓ_1 with prior information.

To illustrate the effects of the representation (especially in this simple dynamics model) before performing a broad comparison across many algorithms, we first run two separate simulations using both the orthonormal Daubechies wavelet transform (DWT) and a four-times overcomplete dual-tree discrete wavelet transform (DT-DWT) [130]. Compared to the DWT, we expect the redundancy in the DT-DWT to produce higher levels of sparsity, which will improve CS recovery overall. Furthermore, we also expect the DT-DWT to be more shift-invariant, leading to better performance of the simple dynamics model that assumes stationary coefficients from frame to frame. We simulate CS measurements by

applying a subsampled noiselet transform [131] to each frame, taking $M = 0.25N$ measurements per frame (where $N = 128^2$) with a measurement noise variance of 10^{-4}.[4] We solve all optimization programs using the TFOCS package [133] due its stability during RWL1 optimization and the ability to use fast implicit operators for matrix multiplications.

In the Foreman video sequence, we simulate CS measurements on a portion comprised of 128×128 pixels. In the DWT recovery we use the following parameters: $\lambda = 0.01$ for BPDN, $\lambda_0 = 0.001$, $\tau = 0.05$ and $\nu = 0.1$ for RWL1, $\gamma = 0.01$ and $\kappa = 0.4$ for BPDN-DF, and $\lambda_0 = 0.001$, $\tau = 0.2$ β, = 1 and $\nu = 0.2$ for RWL1-DF. In the DT-DWT recovery we use the parameters $\lambda = 0.001$ for BPDN, $\lambda_0 = 0.11$, $\tau = 0.2$ and $\nu = 0.1$ for RWL1, $\gamma = 0.01$ and $\kappa =$ 0.2 for BPDN-DF, and $\lambda_0 = 0.003$, $\tau = 4$, $\beta = 1$ and $\nu = 1$ for RWL1-DF. Again, we found these parameter setting through a manual parameter sweep to optimize performance for each algorithm for the number of CS measurements. While the number of measurements was fixed in this simulation for computational tractability, recovery performance could be altered for all algorithms by adjusting the number of CS measurements.

Figure 21 shows the recovery of 200 consecutive frames of video in the Foreman sequence. As expected, we see that all algorithms perform better in the DT-DWT case than the DWT case due to the increased sparsity of the representation (DT-DWT representations used approximately 62% of the coefficients necessary in the DWT). In both cases, RWL1-DF converges to the lowest steady-state rMSE and is able to largely sustain that performance over the sequence. In contrast, BPDN-DF cycles through periods of good performance and poor performance, sometimes performing worse than not using temporal information at all. In essence, BPDN-DF is not robust to model errors, and each time there is motion in the scene (violating the simple dynamics model) the algorithm has to reconverge. The RWL1-DF approach does not exhibit this performance oscillation because the use of second-order statistics to propagate temporal information is less rigid, allowing for more robustness during model errors. As expected, while BPDN-DF is still susceptible

[4] We use the noiselet transform because it can be computed with an efficient implicit transform and has enough similarity to a random measurement that it works well in CS [132].

	BPDN	RWL1	BPDN-DF	RWL1-DF	DCS-AMP	WL1P	modCS
Mean rMSE	3.84%	3.09%	3.29%	**1.63%**	3.48%	3.27%	5.22%
Median rMSE	3.85%	3.07%	2.78%	**1.61%**	2.57%	3.27%	4.58%

Table 1: Mean and median values for compressive recovery of the Foreman video sequence.

to model errors, the fragility in BPDN-DF is somewhat mitigated when using the DT-DWT because the simple model is more accurate in this case.

Figure 22 shows a comparison of recovery for the Foreman video sequence in the DT-DWT basis across several existing algorithms, including DCS-AMP, modCS and WL1P. Again we optimize algorithms parameters manually to achieve the best aggregate performance over the video sequence. To summarize the performance over the entire video sequence, Figure 23 shows histogram plots of the rMSE values over the 200 frames for each recovery process using sparsity in the DT-DWT basis. The mean and median for each histogram are represented by the green dashed line and the red arrow respectively, and are listed in Table 1.

As expected, the algorithms with no temporal information are immune to errors in the dynamic model (since it is not used), reflected in the fact that the mean and median are virtually the same in each of these cases. In contrast, the recovery errors for BPDN-DF are much more spread out, achieving nearly the same median error as the independent algorithms and having a much higher mean error due to the large excursions during model mismatch. In other words, unless a more accurate (and complex) dynamic model can be used, BPDN-DF actually may be a worse choice than not using any temporal information at all. In contrast, RWL1-DF shows a much tighter distribution of errors, having a mean and median significantly lower than alternate approaches. While other algorithms are leveraging temporal information for improved performance over the independent algorithms, RWL1-DF demonstrates significant performance improvements over the alternative approaches in this example.

In addition to the in-depth comparison on the single Foreman video sequence above,

we also perform the same CS recovery task on a database of video sequences from a nature BBC documentary to investigate the performance across a wider range of video characteristics (i.e., including video clips with localized motion and global motion in the scene). We simulated CS measurements for 24 sequences (48-frames each) in the same manner as the Foreman sequence and recovered the frames using the same methodology described above (including parameters optimized during recovery of the Foreman sequence). Figure 24 shows the mean and median *improvement* of RWL1-DF relative to the other algorithms being evaluated. Specifically, the plotted mean improvement is the average of the rMSE difference between RWL1-DF and the comparison algorithm at each frame normalized by the average rMSE for the comparison algorithm across the whole video clip. The median improvement is calculated in the same manner.

The recovery results for this video database show consistent performance improvements for RWL1-DF when evaluated over all video sequences in this database. Additionally, we note that some video sequences were significantly richer in texture and motion than others, resulting in a more challenging recovery task. We identified 13 such video clips that were especially challenging (i.e, those where the average rMSE reconstruction for BPDN is over 1%). For these clips we plot the mean and median percent improvement in Figure 24. RWL1-DF shows very significant improvements within these video sequences, indicating that RWL1-DF is especially beneficial in challenging recovery scenarios.

4.5 Learning Dynamics Functions

As mentioned earlier, one of the main challenges in merging dynamic filtering with sparse signal analysis lies in the statistics of the innovations $\boldsymbol{\nu}_n$ and, thus with the dynamics function $f(\cdot)$. Simple dynamics functions for sparse signals (e.g. the identity function for video sequences used in previous simulations in this chapter) lead to innovations that are highly non-Gaussian and non-stationary. One potential solution we have discussed was to design filtering algorithms robust to the noise terms. In particular, the RWL1-DF algorithm based

on the hierarchical Bayesian model achieves such robustness [13]. More advanced inference models, such as RWL1-DF, however incur additional cost. Even with convergence in a small number of iterations, the total computational cost can still be many times higher than simpler algorithms such as BPDN-DF. As an alternative to addressing the robustness to innovations we can instead address the accuracy of the dynamics model. In particular, we can move from a more complex signal model that needs to be inferred at each time-step to a more complex dynamics model which can be learned *a priori* from exemplar data. As with dictionary learning procedures, in this case the bulk of the additional computation is performed in learning the model, and the inference complexity given the learned model remains mostly unchanged. This section will present a parametrized dynamics model which can be learned in a dictionary learning-type manner. First we will present this model into the simpler BPDN-DF, allowing us to leverage the efficiency and guarantees on BPDN-DF with the benefit of more accurate dynamics. We also present a similar model for the RWL1-DF model, deriving an appropriate EM inference algorithm as well as an alternate learning algorithm.

4.5.1 Learning a Bilinear BPDN-DF Model

To model out dynamics in a cohesive manner, we introduce the parametrized dynamics model in terms of the coefficient vector. Specifically, we model the dynamics as

$$\begin{aligned} \boldsymbol{a}_n &= f(\boldsymbol{a}_{n-1}) + \boldsymbol{\nu} \\ &= \sum_{l=1}^{L} b_n[l] \boldsymbol{F}_l \boldsymbol{a}_{n-1} + \boldsymbol{\nu}, \end{aligned} \tag{55}$$

where $\boldsymbol{\nu}_a$ is the innovations term in the coefficient space. The actual signal $\boldsymbol{x}$ in this model is still defined by the linear generative model $\boldsymbol{x}_n = \boldsymbol{\Psi}\boldsymbol{a}_n$. To perform inference in this model we can make use of the BPDN-DF the cost function where the measurement function is simply the identity function,

$$J(\boldsymbol{a}_n) = \|\boldsymbol{x}_n - \boldsymbol{\Psi}\boldsymbol{a}_n\|_2^2 + \gamma_1 \|\boldsymbol{a}_n\|_1 + \gamma_2 \left\| \boldsymbol{a}_n - \sum_{l=1}^{L} \boldsymbol{F}_l \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1} \right\|_2^2 + \gamma_3 \|\boldsymbol{b}_n\|_1 . \tag{56}$$

This cost function lets us define the contributions of the dynamics functions explicitly. By To get an equivalent gradient descent method for this cost function, we do a similar gradient descent as in most variational methods. With respect to the dictionary $\mathbf{\Psi}$, the cost function is identical (all terms aside from the first term are constant with respect to $\mathbf{\Psi}$). This means that the update rule for $\mathbf{\Psi}$ is

$$\boldsymbol{\psi}_i \leftarrow \mu_\Psi \langle \boldsymbol{a}_n[i](\boldsymbol{x}_n - \mathbf{\Psi}\boldsymbol{a}_n)\rangle.$$

In terms of the dynamics functions, the derivative we need to take to determine the gradient step is

$$\frac{dJ}{d\boldsymbol{F}_k} = d\|\boldsymbol{a}_n - \sum_{l=1}^{L} \boldsymbol{F}_l \boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}\|_2^2,$$

Since the matrix derivative is often simpler to calculate via the element-wise definition, we can express the derivative elements as

$$\begin{aligned}
\frac{dJ}{d\boldsymbol{F}_{l,i,k}} &= \frac{d}{d\boldsymbol{F}_{k,i,k}} \sum_i \left(\boldsymbol{a}_n[i] - \sum_{l=1}^{L}\sum_k \boldsymbol{F}_{l,i,k}\boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}[k]\right)^2 \\
&= -2\boldsymbol{b}_n[l]\left(\boldsymbol{a}_n[i] - \sum_{l=1}^{L}\sum_k \boldsymbol{F}_{l,i,k}\boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}[k]\right)\frac{d}{d\boldsymbol{F}_{k,i,k}}\sum_k \boldsymbol{F}_{l,i,k}\boldsymbol{a}_{n-1}[k] \\
&= -2\boldsymbol{b}_n[l]\left(\boldsymbol{a}_n[i] - \sum_{l=1}^{L}\sum_k \boldsymbol{F}_{l,i,k}\boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}[k]\right)\boldsymbol{a}_{n-1}[k],
\end{aligned}$$

While this learning rule is a little more complicated to derive, the resulting learning rule is simple once found. In fact this learning rule is very similar to the learning rule for the dictionary. We can see this by writing the update rule in matrix form as

$$\boldsymbol{F}_l \leftarrow \mu_F \left\langle \boldsymbol{b}_n[l]\left(\boldsymbol{a}_n - \sum_{l=1}^{L}\boldsymbol{F}_l\boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}\right)\boldsymbol{a}_{n-1}^T\right\rangle.$$

As with the learning step over the dictionary, the learning step over the dynamics is an average of the representation errors weighted by the contributions each dynamics function to that error. The full learning algorithm is outlined in Algorithm 2.

To test the learning Algorithm 2 we designed a series of tests. First we pick the simplest case where there is only a single dynamics function ($L = 1$) and the sparsity dictionary is

Algorithm 2 Coefficient and Dynamics Dictionary Learning Algorithm

Initialize γ, μ_Ψ, μ_F, K, ρ_Ψ, ρ_F
Initialize $\boldsymbol{\Psi}$, $\boldsymbol{F}_l$ as random Gaussian matrices
repeat
  for $k = 1$ to K **do**
    Choose data example $\boldsymbol{x}_n$ uniformly at random
    $\{\widehat{\boldsymbol{a}_n}, \widehat{b_n}\} = \arg\min_{\boldsymbol{a}} \|\boldsymbol{x}_n - \boldsymbol{\Psi}\boldsymbol{a}\|_2^2 + \gamma\|\boldsymbol{a}\|_1 + \gamma_2\|\boldsymbol{a} - \sum_{l=1}^{L} \boldsymbol{F}_l \boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}\|_2^2 + \gamma_3\|\boldsymbol{b}_n\|_1$
    $\Delta\boldsymbol{\psi}_i(k) = \frac{1}{T}\sum_{n=1}^{T} \boldsymbol{a}_n[i]\,(\boldsymbol{x}_n - \boldsymbol{\Psi}\boldsymbol{a}_n)$
    $\Delta\boldsymbol{F}_l(k) = \frac{1}{T}\sum_{n=1}^{T} \boldsymbol{b}_n[l]\left(\boldsymbol{a}_n - \sum_{l=1}^{L} \boldsymbol{F}_l\boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}\right)\boldsymbol{a}_{n-1}$
  end for
  $\boldsymbol{\psi}_i \leftarrow \left[\boldsymbol{\psi}_i + \frac{\mu_\Psi}{K}\sum_k \Delta\boldsymbol{\psi}_i(k)\right]$
  $\boldsymbol{F}_l \leftarrow \left[\boldsymbol{F}_l + \frac{\mu_F}{K}\sum_k \Delta\boldsymbol{F}_l(k)\right]$
  Normalize $\boldsymbol{\psi}_i \leftarrow \boldsymbol{\psi}_i/\|\boldsymbol{\psi}_i\|_2$
  Normalize $\boldsymbol{F}_l \leftarrow \boldsymbol{F}_l/\|\boldsymbol{F}_i\|$
  $\mu_\Psi \leftarrow \rho_\Psi\mu_\Psi$, $\mu_F \leftarrow \rho_F\mu_F$
until Convergence

simply the canonical basis ($\boldsymbol{\Psi} = \boldsymbol{I}$). The single dynamics function was simply a permutation matrix concatenated with a scaling matrix (i.e. signal coefficients could move around as well as be scaled). Figure 25 shows the results of running Algorithm 2 as well as the model used to generate the exemplar data. The learned and true models are a very close qualitative match, differing by a permutation and sign change (the same ambiguity present in all dictionary learning methods). With the success of our method to learn a simple dictionary, we then test the algorithm on another simulated data-set, where instead of a single dynamics function, we simulate twelve distinct permutation and scaling functions, two of which are used at each time-step (i.e. the sparsity of $\boldsymbol{b}_n$ is two). We note that different dynamics can be used at each time step. Figure 26 depicts the results of the learning procedure, showing that again the sparsity dictionary is learned up to a permutation and sign change, and the dynamics functions that are learned are again close qualitative matches to the true dynamics functions.

As a final test, we use the learning algorithm to learn a set of dynamics for natural image patches (i.e. video segments). For images, we learn a size 64 sparsity dictionary for 8x8 image patches concurrently with learning 20 64x64 dynamics functions. Since

no ground truth is available for video sequences, we quantify success by the ability of the learned model to improve performance in an inverse problem setting. The rational behind this method is that better models of data should be able to improve our ability to recover the data from very noisy or incomplete measurements of the data. In keeping with previous sections, we look at compressive recovery of image patches. We take, for each image patch in the video, Gaussian random measurements and corrupt the measurements with white Gaussian noise. The number of measurements was 20% of the size of the image patches. We then recover the image patches using the learned sparsity dictionary in BPDN-DF using both the learned dynamics function as well as a simple identity function in place of the learned dynamics. In the latter case, we are simulating the same BPDN-DF process that was used in previous results in this chapter. In the former case we utilize the learned dynamics to show how the learning allows us to improve reconstruction. Figure 27 depicts the histogram of errors over 100 sequences of 20 8x8 image patches taken at random from a BBC documentary used in [128]. BPDN-DF using the learned dictionary clearly out-performs BPDN-DF with a simple identity function for the dynamics. Specifically, while BPDN-DF using the identity dynamics had the majority of the reconstruction errors clustered about 13% error, using the learned dynamics dropped the majority of these errors, resulting in a median error of 8%. The average improvement we see in using the learned dynamics is 33% improvement in rMSE.

4.5.2 Learning a Bilinear RWL1-DF Model

In Sections 4.3 and 4.5.1 we observed that both more robust algorithms and more accurate models can improve sparse signal tracking performance. The logical extension is to combine both models, deriving a more robust model that can learn and utilize a more accurate signal model. For this task we merge the RWL1-DF algorithm with a similar dynamics model as in Section 4.5.1. As a review, RWL1-DF algorithm has a number of advantages over BPDN-DF. For one, there is no explicit assumption of Gaussian innovations. In

fact there are no explicit assumptions on the innovations at all. This fact allows RWL1-DF to be more adaptable when the innovations contains large, yet concentrated, energy. Additionally, the RWL1-DF algorithm does not in any way directly enforce the dynamic information, instead using the prediction from the previous state to encourage certain support sets, thereby resulting in a more reliably sparse solution. The algorithm is at the heart an expectation-maximization algorithm, necessitating an iterative procedure where each iteration requires solving a BPDN-type optimization program. On the other hand, the dictionary learning procedure derived for BPDN-DF incurs no significant additional inference cost, however requires a potentially computationally-intensive learning procedure be done prior to inference. To merge these two methods we use the hierarchical model as in RWL1-DF, however we replace the assumed dynamics with the linear dynamics model from Section 4.5.1. The following sections outline the resulting mathematical model, as well as the derived inference and learning rules.

4.5.2.1 RWL1-DF Bilinear Cost Function

As the EM procedure and the resulting re-weighted algorithm is more complex, we note that the actual cost function is rather complicated. The true cost function being optimized, including the linear mixture model for the dynamics, is

$$J(\boldsymbol{a}_n) = \|\boldsymbol{x}_n - \boldsymbol{\Psi}\boldsymbol{a}_n\|_2^2 - 2\sigma^2 \sum_{k=1}^{N} \log\left(\frac{\alpha\xi(|\left[\sum_l \boldsymbol{F}_l \boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}\right]_k| + \eta)^{-1}}{2\left(\xi\lambda_0 \frac{|\boldsymbol{a}_n[k]|}{|\left[\sum_l \boldsymbol{F}_l \boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}\right]_k|+\eta} + 1\right)^{\alpha+1}}\right).$$

We can simplify this expression by separated over the logarithms and removing all

constants that do not effect the actual optimization solution as

$$\begin{aligned} J(\boldsymbol{a}_n) &= \|\boldsymbol{x}_n - \boldsymbol{\Psi}\boldsymbol{a}_n\|_2^2 + 2\sigma^2\left(\sum_{k=1}^{N}\log\left(|\left[\sum_l \boldsymbol{F}_l\boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}\right]_k| + \eta\right)\right. \\ &\quad \left. -N\log(\alpha\xi) + (\alpha+1)\sum_{k=1}^{N}\log\left(\xi\lambda_0\frac{|\boldsymbol{a}_n[k]|}{|\left[\sum_l \boldsymbol{F}_l\boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}\right]_k| + \eta} + 1\right) + N\log(2)\right) \\ &= \|\boldsymbol{x}_n - \boldsymbol{\Psi}\boldsymbol{a}_n\|_2^2 + 2\sigma^2\left(\sum_{k=1}^{N}\log\left(\left|\left[\sum_l \boldsymbol{F}_l\boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}\right]_k\right| + \eta\right)\right. \\ &\quad \left. +(\alpha+1)\sum_{k=1}^{N}\log\left(\xi\lambda_0\frac{|\boldsymbol{a}_n[k]|}{|\left[\sum_l \boldsymbol{F}_l\boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}\right]_k| + \eta} + 1\right)\right). \end{aligned}$$

For further clarity, we can further simplify this expression as

$$\begin{aligned} J(\boldsymbol{a}_n) &= \|\boldsymbol{x}_n - \boldsymbol{\Psi}\boldsymbol{a}_n\|_2^2 + 2\sigma^2\left(\sum_{k=1}^{N}\log\left(\left|\left[\sum_l \boldsymbol{F}_l\boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}\right]_k\right| + \eta\right)\right. \\ &\quad \left. +(\alpha+1)\sum_{k=1}^{N}\log\left(\frac{\xi\lambda_0|\boldsymbol{a}_n[k]| + \left|\left[\sum_l \boldsymbol{F}_l\boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}\right]_k\right| + \eta}{\left|\left[\sum_l \boldsymbol{F}_l\boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}\right]_k\right| + \eta}\right)\right) \\ &= \|\boldsymbol{x}_n - \boldsymbol{\Psi}\boldsymbol{a}_n\|_2^2 + 2\sigma^2\left(\sum_{k=1}^{N}\log\left(\left|\left[\sum_l \boldsymbol{F}_l\boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}\right]_k\right| + \eta\right) + \right. \\ &\quad +(\alpha+1)\sum_{k=1}^{N}\log\left(\xi\lambda_0|\boldsymbol{a}_n[k]| + \left|\left[\sum_l \boldsymbol{b}_n[l]\boldsymbol{F}\boldsymbol{a}_{n-1}\right]_k\right| + \eta\right) \\ &\quad \left. + -(\alpha+1)\sum_{k=1}^{N}\log\left(\left|\left[\sum_l \boldsymbol{F}_l\boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}\right]_k\right| + \eta\right)\right) \\ &= \|\boldsymbol{x}_n - \boldsymbol{\Psi}\boldsymbol{a}_n\|_2^2 + 2\sigma^2\left((\alpha+1)\sum_{k=1}^{N}\log\left(\xi\lambda_0|\boldsymbol{a}_n[k]| + \left|\left[\sum_l \boldsymbol{F}_l\boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}\right]_k\right| + \eta\right) - \right. \\ &\quad \left. \alpha\sum_{k=1}^{N}\log\left(\left|\left[\sum_l \boldsymbol{F}_l\boldsymbol{b}_n[l]\boldsymbol{a}_{n-1}\right]_k\right| + \eta\right)\right). \end{aligned} \tag{57}$$

We can see from the final expression in Equation (57) that the ideal cost function to optimize comes in three parts. The first term is the traditional measurement fidelity term. The second term is essentially the same sparsity-inducing cost seen in the ideal RWL1-DF cost function. The final term penalizes the values of the dynamics coefficients. Balancing these three terms is exceedingly difficult, especially since two of the terms are not convex. To obtain appropriate inference procedures, in addition to learning rules to optimize $\boldsymbol{\Psi}$ and $\boldsymbol{F}_l$ we again derive an EM algorithm based on a hierarchical model.

4.5.2.2 *Inference Rule*

Recall that in the RWL1-DF algorithm, where the dynamics function is known (in the bilinear case, this means that the $\boldsymbol{b}$'s are known), the inference via the EM algorithm is

$$\begin{aligned}\widehat{\boldsymbol{a}}_n^t &= \arg\min_{\boldsymbol{a}} \|\boldsymbol{x}_n - \boldsymbol{\Psi}\boldsymbol{a}\|_2^2 + \lambda_0 \sum_k \lambda_n^{t-1}[k]\,|\boldsymbol{a}[k]| \\ \boldsymbol{\lambda}_n^t[k] &= \frac{2(\alpha+1)\xi}{\lambda_0\xi\left|\widehat{\boldsymbol{a}}_n^t[k]\right| + \left|[f(\widehat{\boldsymbol{a}}_{n-1})]_k\right| + \eta}.\end{aligned}$$

To extend this model and to derive a similar iterative algorithm for the bilinear model, we first note that inferring $\boldsymbol{a}$ conditioned on $\boldsymbol{b}$ reduces to the same RWL1 process, only with the update on $\boldsymbol{\lambda}$ replaced by

$$\boldsymbol{\lambda}_n^t[k] = \frac{2(\alpha+1)\xi}{\lambda_0\xi\left|\widehat{\boldsymbol{a}}_n^t[k]\right| + \left|[\sum_l \boldsymbol{F}_l \boldsymbol{b}_n[l]\widehat{\boldsymbol{a}}_{n-1}]_k\right| + \eta}$$

Inferring $\boldsymbol{b}$, however, is a more involved process. We can, as in the BPDN-DF case, provide a prior distribution over the dynamics coefficients as well. To retain sparsity, we give $\boldsymbol{b}$ the same distribution as $\boldsymbol{a}$. As for $\boldsymbol{a}$, we again describe the prior over $\boldsymbol{b}$ as a conditional prior in terms of a set of latent variables $\boldsymbol{\gamma}$,

$$p(\boldsymbol{b}|\boldsymbol{\gamma}) = \prod_l \frac{\gamma_0\boldsymbol{\gamma}_l}{2} e^{-\gamma_0\boldsymbol{\gamma}_l|\boldsymbol{b}[l]|}$$

where the latent variables γ have a conjugate distribution

$$p(\boldsymbol{\gamma}) = \prod_l \frac{\boldsymbol{\gamma}_l^{\tau+1}}{\theta^\tau\Gamma(\tau)} e^{-\boldsymbol{\gamma}_l/\theta}$$

The complete maximum a-posteriori estimate should then be

$$\begin{aligned}\{\widehat{\boldsymbol{a}},\widehat{\boldsymbol{b}}\} &= \arg\max_{\boldsymbol{a},\boldsymbol{b}} p(\boldsymbol{a},\boldsymbol{b}|\boldsymbol{x}) \\ &= \arg\max_{\boldsymbol{a},\boldsymbol{b}} \int_{\boldsymbol{\gamma}}\int_{\boldsymbol{\lambda}} p(\boldsymbol{a},\boldsymbol{b},\boldsymbol{\lambda},\boldsymbol{\gamma}|\boldsymbol{a},\boldsymbol{b})d\boldsymbol{\lambda}d\boldsymbol{\gamma} \\ &= \arg\max_{\boldsymbol{a},\boldsymbol{b}} \int_{\boldsymbol{\gamma}}\int_{\boldsymbol{\lambda}} p(\boldsymbol{x}|\boldsymbol{a},\boldsymbol{b},\boldsymbol{\lambda},\boldsymbol{\gamma})p(\boldsymbol{a},\boldsymbol{b},\boldsymbol{\gamma},\boldsymbol{\lambda})d\boldsymbol{\lambda}d\boldsymbol{\gamma} \\ &= \arg\max_{\boldsymbol{a},\boldsymbol{b}} \int_{\boldsymbol{\gamma}}\int_{\boldsymbol{\lambda}} p(\boldsymbol{x}|\boldsymbol{a})p(\boldsymbol{a}|\boldsymbol{\lambda})p(\boldsymbol{\lambda}|\boldsymbol{b})p(\boldsymbol{b}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})d\boldsymbol{\lambda}d\boldsymbol{\gamma}\end{aligned}$$

Given the dependence of the latent variables $\boldsymbol{\gamma}$ and $\boldsymbol{\lambda}$, we can use an EM algorithm again, separating the problem into the two steps

$$\text{M-step:} \qquad \{\widehat{\boldsymbol{a}}, \widehat{\boldsymbol{b}}\} \arg\max_{\boldsymbol{a},\boldsymbol{b}} p(\boldsymbol{a}, \boldsymbol{b}|\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\gamma}) \tag{58}$$

$$\text{E-step:} \qquad \{\widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\gamma}}\} = \{E_{p(\boldsymbol{\lambda},\boldsymbol{\gamma}|\boldsymbol{b},\boldsymbol{a},\boldsymbol{x})}[\boldsymbol{\lambda}], E_{p(\boldsymbol{\lambda},\boldsymbol{\gamma}|\boldsymbol{b},\boldsymbol{a},\boldsymbol{x})}[\boldsymbol{\gamma}]\} \tag{59}$$

For the M-step, we can easily write

$$\begin{aligned}
\{\widehat{\boldsymbol{a}}, \widehat{\boldsymbol{b}}\} &= \arg\max_{\boldsymbol{a},\boldsymbol{b}} p(\boldsymbol{a}, \boldsymbol{b}|\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\gamma}) \\
&= \arg\max_{\boldsymbol{a},\boldsymbol{b}} p(\boldsymbol{x}|\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\lambda}, \boldsymbol{\gamma}) p(\boldsymbol{a}, \boldsymbol{b}|\boldsymbol{\lambda}, \boldsymbol{\gamma}) \\
&= \arg\max_{\boldsymbol{a},\boldsymbol{b}} p(\boldsymbol{x}|\boldsymbol{a}) p(\boldsymbol{a}|\boldsymbol{\lambda}) p(\boldsymbol{b}|\boldsymbol{\lambda}, \boldsymbol{\gamma}) \\
&= \arg\max_{\boldsymbol{a},\boldsymbol{b}} p(\boldsymbol{x}|\boldsymbol{a}) p(\boldsymbol{a}|\boldsymbol{\lambda}) p(\boldsymbol{\lambda}|\boldsymbol{b}) p(\boldsymbol{b}|\boldsymbol{\gamma}),
\end{aligned}$$

where the third step follows from the conditional independence between $\boldsymbol{a}$ and $\boldsymbol{b}$. Since the maximum of the product of two functions is the product of their maxima, this optimization can easily be split into two parts:

$$\begin{aligned}
\widehat{\boldsymbol{a}} &= \arg\max_{\boldsymbol{a}} p(\boldsymbol{x}|\boldsymbol{a}) p(\boldsymbol{a}|\boldsymbol{\lambda}) \\
\widehat{\boldsymbol{b}} &= \arg\max_{\boldsymbol{b}} p(\boldsymbol{\lambda}|\boldsymbol{b}) p(\boldsymbol{b}|\boldsymbol{\gamma}).
\end{aligned}$$

The first of these two optimizations is again (after applying the negative log function) a weighted BPDN optimization:

$$\widehat{\boldsymbol{a}}_n = \arg\min_{\boldsymbol{a}} \frac{1}{2\sigma^2} \|\boldsymbol{x}_n - \boldsymbol{\Psi}\boldsymbol{a}\|_2^2 + \lambda_0 \sum_i \widehat{\boldsymbol{\lambda}}_n[i]|\boldsymbol{a}[i]|. \tag{60}$$

The second optimization is more involved. The conditional probability of the coefficient variances based on the dynamics coefficients can be derived as

$$p(\boldsymbol{\lambda}_n|\boldsymbol{b}_n) = \prod_i \frac{\boldsymbol{\lambda}_n^{\alpha+1}[i] \left(\left| [\sum_l \boldsymbol{F}_l \boldsymbol{b}_n[l] \widehat{a}_{n-1}]_i \right| + \eta \right)^{\alpha}}{\xi^{\alpha} \Gamma(\tau)} e^{-\boldsymbol{\lambda}_i (\left| [\sum_l \boldsymbol{F}_l \boldsymbol{b}_n[l] \widehat{a}_{n-1}]_i \right| + \eta)/\xi}.$$

To simplify this expression, we can define the vector

$$\boldsymbol{u}_{n,i} = \left[[\boldsymbol{F}_1 \widehat{a}_{n-1}]_i, [\boldsymbol{F}_2 \widehat{a}_{n-1}]_i, \ldots, [\boldsymbol{F}_L \widehat{a}_{n-1}]_i \right]^T,$$

which is essentially the vector made up the i^{th} coefficient predictions from all L dynamics functions. The simplified conditional probability is then

$$p(\boldsymbol{\lambda}_n|\boldsymbol{b}_n) = \prod_i \frac{\boldsymbol{\lambda}_n^{\alpha+1}[i]\left(\left|\langle \boldsymbol{u}_{n,i}, \boldsymbol{b}_n\rangle\right| + \eta\right)^{\alpha}}{\xi^{\alpha}\Gamma(\tau)} e^{-\boldsymbol{\lambda}_n[i](\left|\langle \boldsymbol{u}_{n,i}, \boldsymbol{b}_n\rangle\right|+\eta)/\xi}.$$

The product with the conditional probability over the dynamics coefficients is

$$p(\boldsymbol{\lambda}_n|\boldsymbol{b}_n)p(\boldsymbol{b}_n|\boldsymbol{\gamma}_n) = \left(\prod_l \frac{\gamma_0\boldsymbol{\gamma}_n[l]}{2}\right)\left(\prod_i \frac{\boldsymbol{\lambda}_n^{\alpha+1}[i]\left(\left|\langle \boldsymbol{u}_{n,i}, \boldsymbol{b}_n\rangle\right| + \eta\right)^{\alpha}}{\xi^{\alpha}\Gamma(\tau)}\right) e^{-\frac{1}{\xi}\sum_i \boldsymbol{\lambda}_n[i](\left|\langle \boldsymbol{u}_{n,i}, \boldsymbol{b}_n\rangle\right|+\eta) - \gamma_0 \sum_l \boldsymbol{\gamma}_n[l]|\boldsymbol{b}_n[l]|}$$

Taking the negative logarithm of this expression, and removing all terms constant with respect to $\boldsymbol{b}$, we can see that the MAP optimization is equivalent to

$$\arg\min_{\boldsymbol{b}} \frac{1}{\xi}\sum_i \boldsymbol{\lambda}_n[i]\left|\langle \boldsymbol{u}_{n,i}, \boldsymbol{b}_n\rangle\right| + \gamma_0 \sum_l \boldsymbol{\gamma}_l|\boldsymbol{b}_n[l]| - \alpha \sum_i \log\left(\left|\langle \boldsymbol{u}_{n,i}, \boldsymbol{b}_n\rangle\right| + \eta\right)$$

By letting

$$\boldsymbol{U}_n = \left[\boldsymbol{F}_1\widehat{a}_{n-1}, \boldsymbol{F}_2\widehat{a}_{n-1}, \ldots, \boldsymbol{F}_L\widehat{a}_{n-1}\right]^T,$$

and $\boldsymbol{\Gamma} = \mathrm{diag}(\boldsymbol{\gamma}_n)$, we can write this second optimization concisely as

$$\arg\min_{\boldsymbol{b}} \left\| \begin{bmatrix} \frac{1}{\alpha\xi}\boldsymbol{\Lambda}\boldsymbol{U}_n \\ \frac{\gamma_0}{\alpha}\boldsymbol{\Gamma}_n \end{bmatrix} \boldsymbol{b}_n \right\|_1 - \sum_i \log\left(\left|\langle \boldsymbol{u}_{n,i}, \boldsymbol{b}_n\rangle\right| + \eta\right) \tag{61}$$

This optimization program is not necessarily convex, however it seems quasi-convex.

For the E-step, We can likewise split the expectation into finding the expectation of two independent sets of random variables:

$$\begin{aligned} \widehat{\boldsymbol{\lambda}}_n &= E_{p(\boldsymbol{\lambda}_n, \boldsymbol{\gamma}_n|\boldsymbol{b}_n, \boldsymbol{a}_n, \boldsymbol{x}_n)}[\boldsymbol{\lambda}] \\ \widehat{\boldsymbol{\gamma}}_n &= E_{p(\boldsymbol{\lambda}_n, \boldsymbol{\gamma}_n|\boldsymbol{b}_n, \boldsymbol{a}_n, \boldsymbol{x}_n)}[\boldsymbol{\gamma}]. \end{aligned}$$

Thanks to the conjugacy of the Laplacian and Gamma distributions, both expectations have closed form solutions, giving the updates

$$\begin{aligned} \widehat{\boldsymbol{\lambda}}_n[i] &= \frac{(\alpha+1)\xi}{\xi\lambda_0|\widehat{\boldsymbol{a}}_n[i]| + \left|\left[\sum_l \boldsymbol{F}_l\widehat{\boldsymbol{b}}_n[l]\widehat{\boldsymbol{a}}_{n-1}\right]_i\right| + \eta} \\ \widehat{\boldsymbol{\gamma}}_n[l] &= \frac{(\tau+1)\theta}{\theta\gamma_0|\widehat{\boldsymbol{b}}_n[l]| + 1}. \end{aligned}$$

To summarize, the algorithm alternates between two steps, an update of $\{\widehat{\boldsymbol{a}}_n, \widehat{\boldsymbol{b}}_n\}$ and an update of $\{\widehat{\boldsymbol{\lambda}}_n, \widehat{\boldsymbol{\gamma}}_n\}$:

M-step:

$$\widehat{\boldsymbol{a}}_n = \arg\min_{\boldsymbol{a}} \frac{1}{2\sigma^2}\|\boldsymbol{x}_n - \boldsymbol{\Psi}\boldsymbol{a}\|_2^2 + \lambda_0\|\widehat{\boldsymbol{\Lambda}}_n\boldsymbol{a}\|_1 \tag{62}$$

$$\widehat{\boldsymbol{b}}_n = \arg\min_{\boldsymbol{b}} \left\| \begin{bmatrix} \frac{1}{\alpha\xi}\widehat{\boldsymbol{\Lambda}}\boldsymbol{U}_n \\ \frac{\gamma_0}{\alpha}\widehat{\boldsymbol{\Gamma}}_n \end{bmatrix} \boldsymbol{b} \right\|_1 - \sum_i \log\left(\left|\langle \boldsymbol{u}_{n,i}, \boldsymbol{b}_n\rangle\right| + \eta\right) \tag{63}$$

E-step:

$$\widehat{\boldsymbol{\lambda}}_n[i] = \frac{(\alpha+1)\xi}{\xi\lambda_0|\widehat{\boldsymbol{a}}_n[i]| + \left|\langle \boldsymbol{u}_{n,i}\widehat{\boldsymbol{b}}_n\rangle\right| + \eta} \tag{64}$$

$$\widehat{\boldsymbol{\gamma}}_n[l] = \frac{(\tau+1)\theta}{\theta\gamma_0|\widehat{\boldsymbol{b}}_n[l]| + 1}. \tag{65}$$

In the M-step, we define $\widehat{\boldsymbol{\Lambda}} = \mathrm{diag}(\widehat{\boldsymbol{\lambda}})$ and again we define $\widehat{\boldsymbol{\Gamma}} = \mathrm{diag}(\widehat{\boldsymbol{\gamma}})$.

As mentioned before, most of these steps are easy to solve. In particular, The updates in Equations (64) and (65) are simple closed-form solutions and easy to evaluate. The optimization program in Equation (62) is simply a weighted ℓ_1 regularized least-squares optimization for which many fast solvers exist. The difficult optimization is Equation (63), which is convex over each of $2^L + L(N-L)$ (since $L < N$ distinct regions defined by the hyperplanes that have the $\boldsymbol{u}_{n,i}$ vectors as their normals. Since we can plainly see that the cost function evaluated at $\boldsymbol{b}$ is identical to the cost function evaluated at $-\boldsymbol{b}$ we can restrict the number of regions of interest to $2^{L-1} + L(N-L)/2$ regions. To find the global optimum, the region containing the global minimum needs to be determined. Once that region is determined, the problem reduces to a convex optimization (optimization of a convex function over a cone).

While locating the correct region would normally require $2^{L-1}+L(N-L)/2$ optimization problems be solved in parallel, and the answers then compared, we can use the ℓ_1 norm and the behavior of the sum-log function in each region to find a heuristic to choose a single (or small number of) regions to optimize over. In particular we can use the radial symmetry of

the sum-log function to sample each region once, using the single sample in conjunction with the $\widehat{\boldsymbol{\lambda}}$ and $\widehat{\boldsymbol{\gamma}}$ values to test which regions are more or less likely to have the global minimum.

4.5.2.3 Learning Rule

In the RWL1-DF model, the learning rule for the dictionary $\boldsymbol{\Psi}$ remains the same as in previous models. This fact is due to all terms in the cost function, aside from the measurement fidelity term, are not dependent on the sparsity-inducing dictionary. It remains only to calculate the learning rule for the dynamics dictionary. To calculate the learning rule for $\boldsymbol{F}_l$, we again need to calculate calculate the derivative of the cost function of Equation (57) with respect to the dynamics matrices:

$$\frac{dJ}{d\boldsymbol{F}_{l,k,i}} = 2\sigma^2 \frac{d}{d\boldsymbol{F}_{l,k,i}} \left((\alpha+1) \sum_{k=1}^{N} \log\left(\xi\lambda_0 |\boldsymbol{a}_n[k]| + \left| \left[\sum_l \boldsymbol{F}_l \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1} \right]_k \right| + \eta \right) \right.$$
$$\left. -\alpha \sum_{k=1}^{N} \log\left(\left(\left| \left[\sum_l \boldsymbol{F}_l \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1} \right]_k \right| + \eta \right) \right) \right).$$

To simplify a bit, we note that $[\boldsymbol{F}\boldsymbol{a}_{n-1}]_k = \sum_i \boldsymbol{F}_{k,i} \boldsymbol{a}_{n-1}[i]$, or, in the bilinear model we obtain $\left[\sum_l \boldsymbol{F}_l \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1}\right]_k = \sum_{i,l} \boldsymbol{F}_{l,k,i} \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1}[i]$. We can use this to calculate the actual derivative

$$\begin{aligned} \frac{dJ}{d\boldsymbol{F}_{l,k,i}} &= 2\sigma^2(\alpha+1) \frac{d}{d\boldsymbol{F}_{l,k,i}} \log\left(\xi\lambda_0 |\boldsymbol{a}_n[k]| + \left| \sum_{i,l} \boldsymbol{F}_{l,k,i} \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1}[i] \right| + \eta \right) \\ &\quad -2\sigma^2\alpha \frac{d}{d\boldsymbol{F}_{l,k,i}} \log\left(\left| \sum_{i,l} \boldsymbol{F}_{l,k,i} \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1}[i] \right| + \eta \right). \end{aligned}$$

This can be calculated as

$$\frac{dJ}{d\boldsymbol{F}_{l,k,i}} = \frac{2\sigma^2(\alpha+1) \frac{d}{d\boldsymbol{F}_{l,k,i}} |\sum_{i,l} \boldsymbol{F}_{l,k,i} \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1}[i]|}{\xi\lambda_0 |\boldsymbol{a}_n[k]| + |\sum_{i,l} \boldsymbol{F}_{l,k,i} \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1}[i]| + \eta} - \frac{2\sigma^2\alpha \frac{d}{d\boldsymbol{F}_{l,k,i}} |\sum_{i,l} \boldsymbol{F}_{l,k,i} \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1}[i]|}{|\sum_{i,l} \boldsymbol{F}_{l,k,i} \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1}[i]| + \eta}$$

$$\frac{dJ}{d\boldsymbol{F}_{l,k,i}} = \frac{2\sigma^2(|\sum_{i,l} \boldsymbol{F}_{l,k,i} \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1}[i]| + \eta - \alpha\xi\lambda_0 |\boldsymbol{a}_n[k]|)}{(\xi\lambda_0 |\boldsymbol{a}_n[k]| + |\sum_{i,l} \boldsymbol{F}_{l,k,i} \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1}[i]| + \eta)(|\sum_{i,l} \boldsymbol{F}_{l,k,i} \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1}[i]| + \eta)} \frac{d}{d\boldsymbol{F}_{l,k,i}} |\sum_{i,l} \boldsymbol{F}_{l,k,i} \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1}[i]|$$

$$\frac{dJ}{d\boldsymbol{F}_{l,k,i}} = \frac{2\sigma^2(|\sum_{i,l} \boldsymbol{F}_{l,k,i} \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1}[i]| + \eta - \alpha\xi\lambda_0 |\boldsymbol{a}_n[k]|) \text{sign}\left(\sum_{i,l} \boldsymbol{F}_{l,k,i} \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1}[i]\right) \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1}[i]}{(\xi\lambda_0 |\boldsymbol{a}_n[k]| + |\sum_{i,l} \boldsymbol{F}_{l,k,i} \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1}[i]| + \eta)(|\sum_{i,l} \boldsymbol{F}_{l,k,i} \boldsymbol{b}_n[l] \boldsymbol{a}_{n-1}[i]| + \eta)}$$

Which gives the gradient-step update for the dictionary elements.

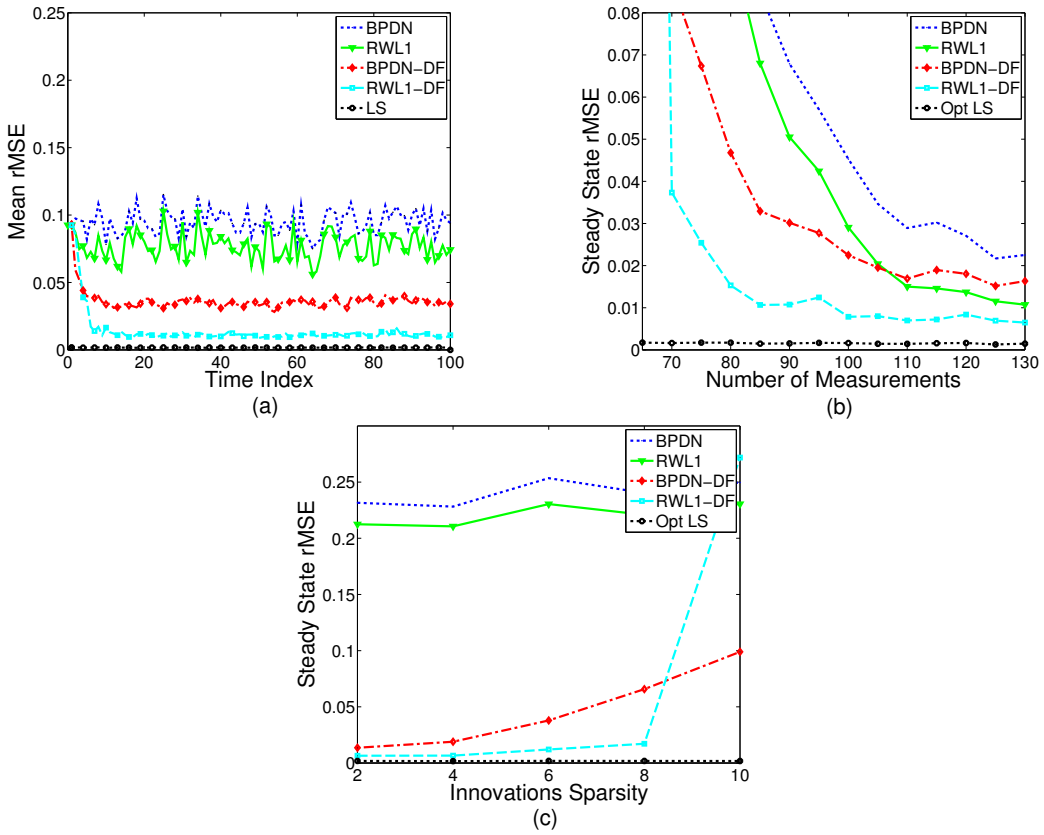

Figure 20: Behavior of the RWL1-DF algorithm on synthetic data. (a) RWL1-DF converges to a lower mean rMSE than static sparse estimation or BPDN-DF. Shown for $M = 80$, $N = 576$, $S = 20$ and $p = 0.25$. (b) When sweeping the number of measurements M for $N = 576$, $S = 20$, and $p = 0.25$, we observe that the performance improvement for RWL1-DF is especially distinct in the highly undersampled regime. Each point is the average steady state rMSE over 40 independent trials. (c) RWL1-DF is also more robust to model mismatch in the innovation statistics. Shown here for different innovations sparsity ($2Sp$) for $M = 70$, $N = 576$, and $S = 20$. Each data point is the result of averaging the steady-state rMSE over 40 independent trials. Note that when BPDN-DF starts to perform better ($2Sp = 10$), the innovations are actually half of the total support set and a Gaussian innovations model may be more accurate than a sparse innovation model.

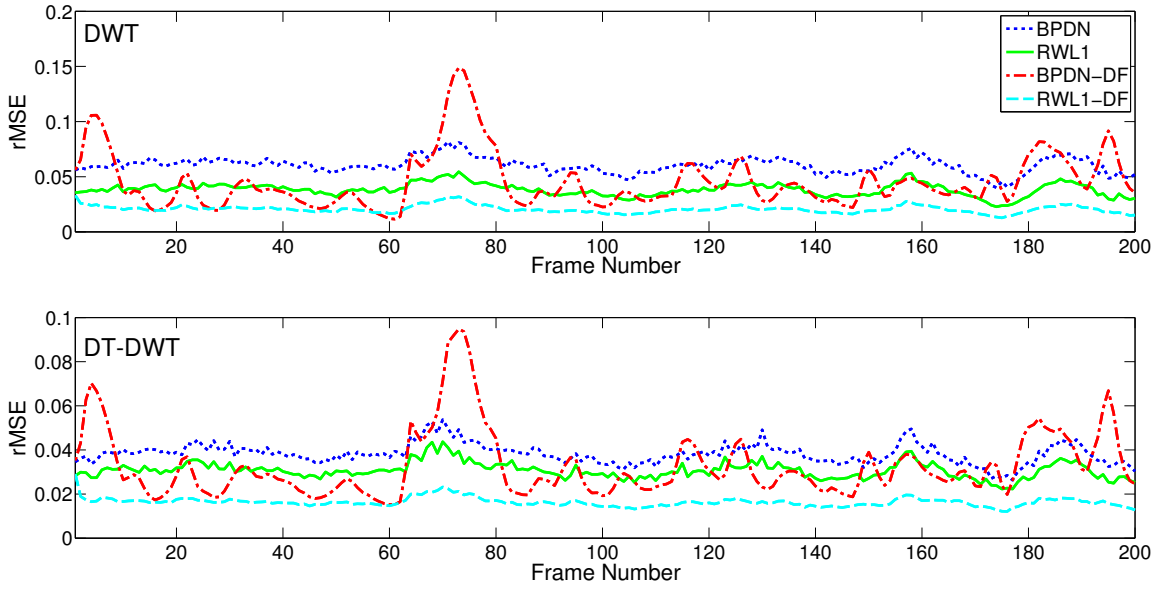

Figure 21: CS recovery of the full Foreman video sequence. Each curve represents the rMSE for recovery from subsampled noiselets ($M/N = 0.25$) using either a DWT (top) or a DT-DWT (bottom) as the sparsifying basis. The independent BPDN recovery (dotted blue curve) and the independent re-weighted BPDN (solid green curve) retain a steady rMSE over time. RWL1-DF (dashed cyan curve) converges on a lower rMSE than either time-independent estimation and remains at approximately steady-state for the remainder of the video sequence. BPDN-DF (the dot-dash red curve) can converge to low rMSE values, but is highly unstable and can yield very poor results when the model is not accurate due to motion in the scene. Compared to using the four-times overcomplete DT-DWT, when the orthonormal DWT is used as the sparsifying basis all algorithms except RWL1-DF (but especially BPDN-DF) suffer in performance due to the dynamic signal model being less accurate.

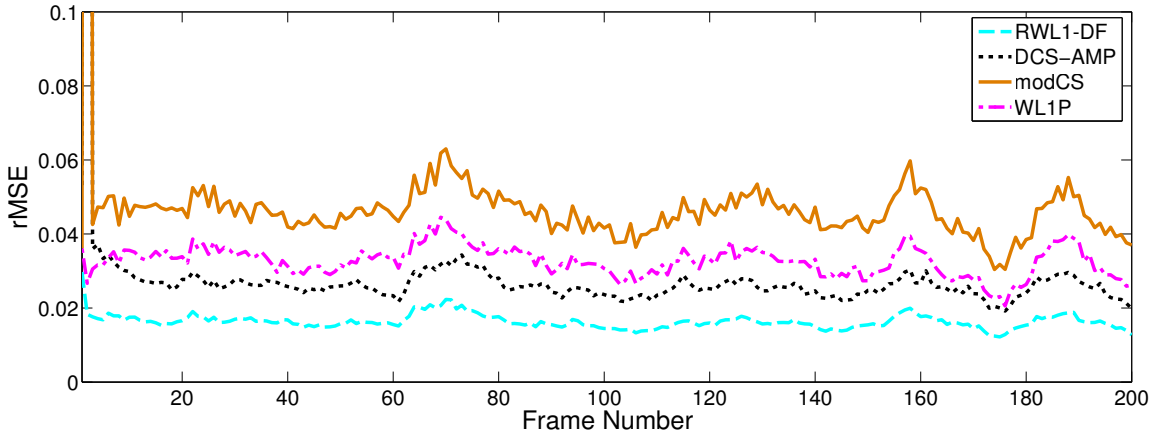

Figure 22: A comparison of RWL1-DF with existing recovery algorithms (DCS-AMP, modCS and WL1P) for the Foreman video sequence. Each curve represents the rMSE for recovery from subsampled noiselets ($M/N = 0.25$) using the DT-DWT as the sparsifying basis.

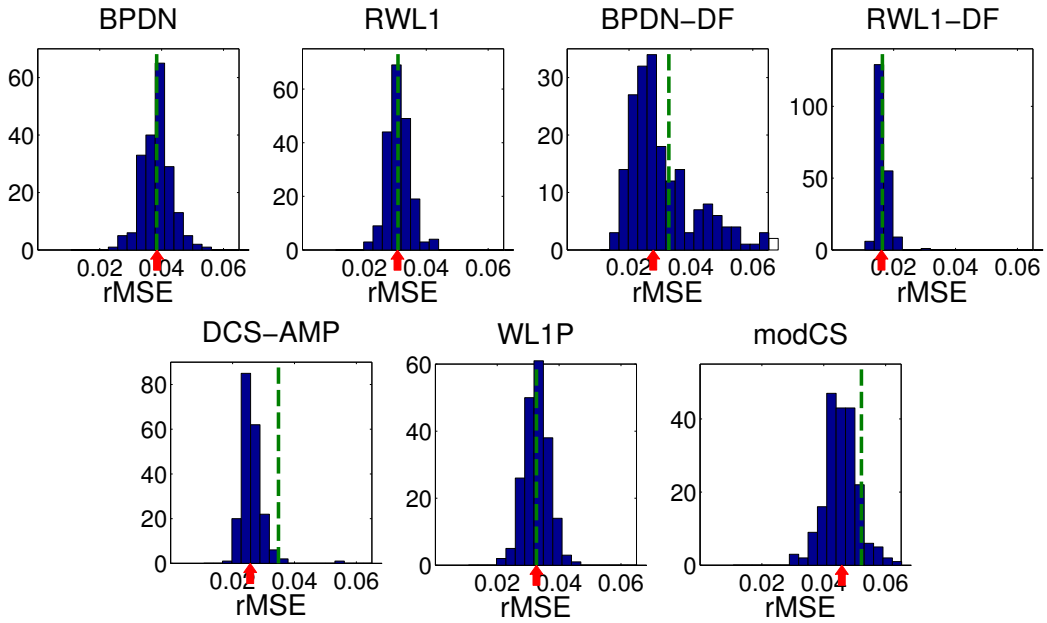

Figure 23: Histogram of the rMSE for the compared algorithms when recovering the Foreman video sequence with the DT-DWT as the sparsifying basis. RWL1-DF achieves a lower mean (indicated by the dashed green lines) and median (indicated by the red arrows), with a tightly concentrated error distribution due to the robustness to model mismatch (producing few outliers). Specific mean and median values are shown in Table 1.

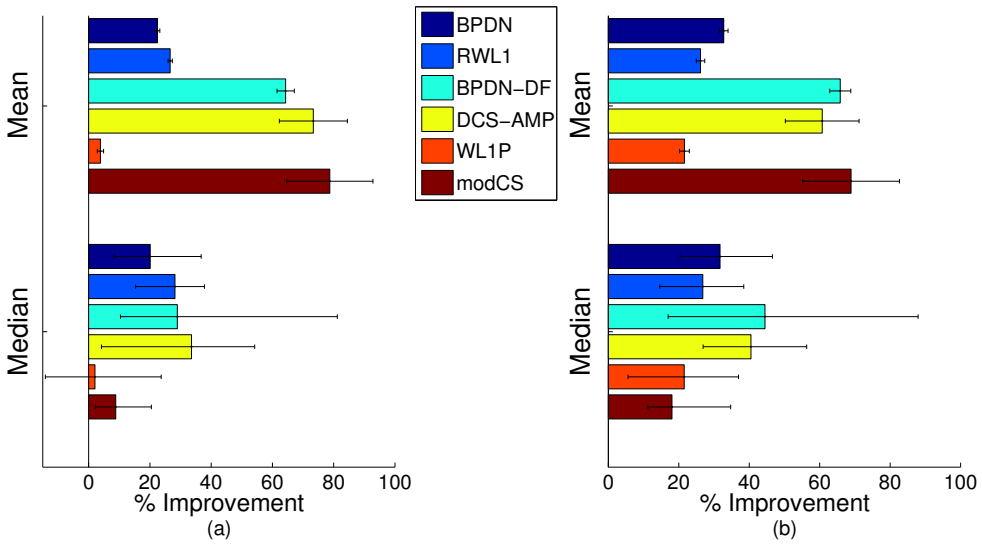

Figure 24: Percent improvement of RWL1-DF over other algorithms for compressive recovery of video sequences (calculated as described in the text). Displayed is both the mean improvement (error bars indicating the normalized standard deviation) and the median improvement (error bars indicating the 25th and 75th percentile) for each algorithm. (a) Results for the full database of 24 sequences from a BBC nature documentary. (b) Results for the 13 sequences that were especially challenging for CS recovery, illustrating the benefits of RWL1-DF in this particularly difficult regime.

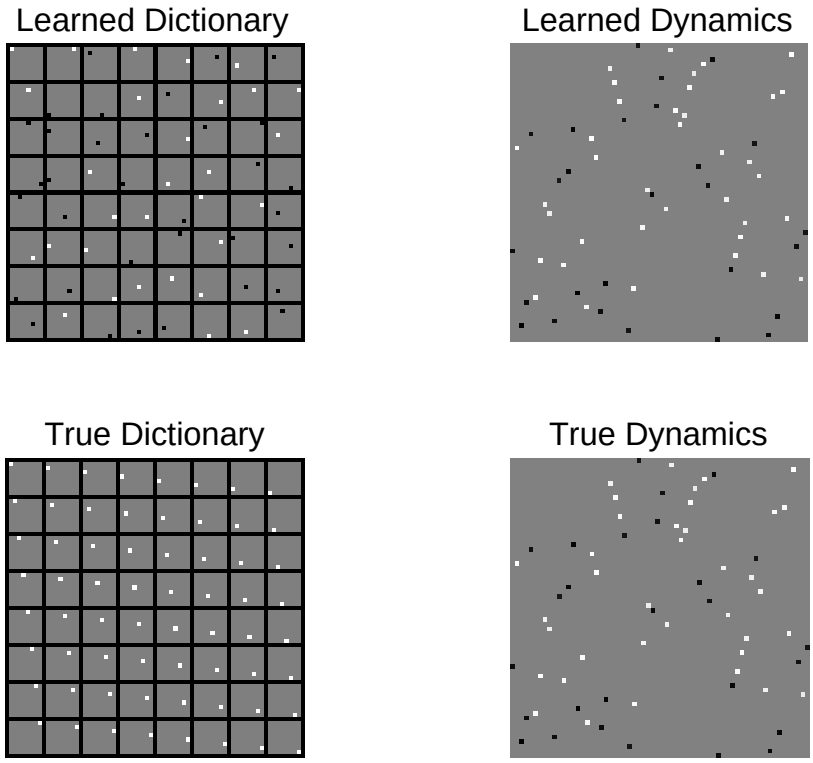

Figure 25: The dynamics learning algorithm can learn an identity basis and a simple permutation dynamics function.

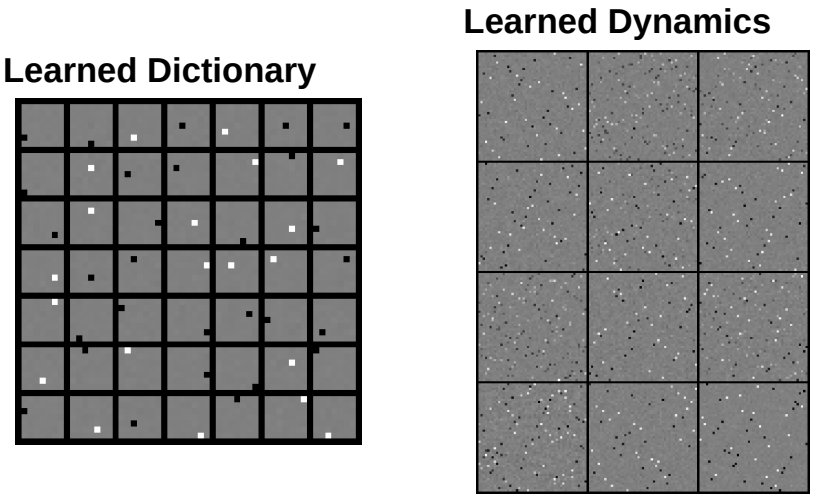

Figure 26: The dynamics learning algorithm can learn an identity basis and a set of permutation dynamics functions.

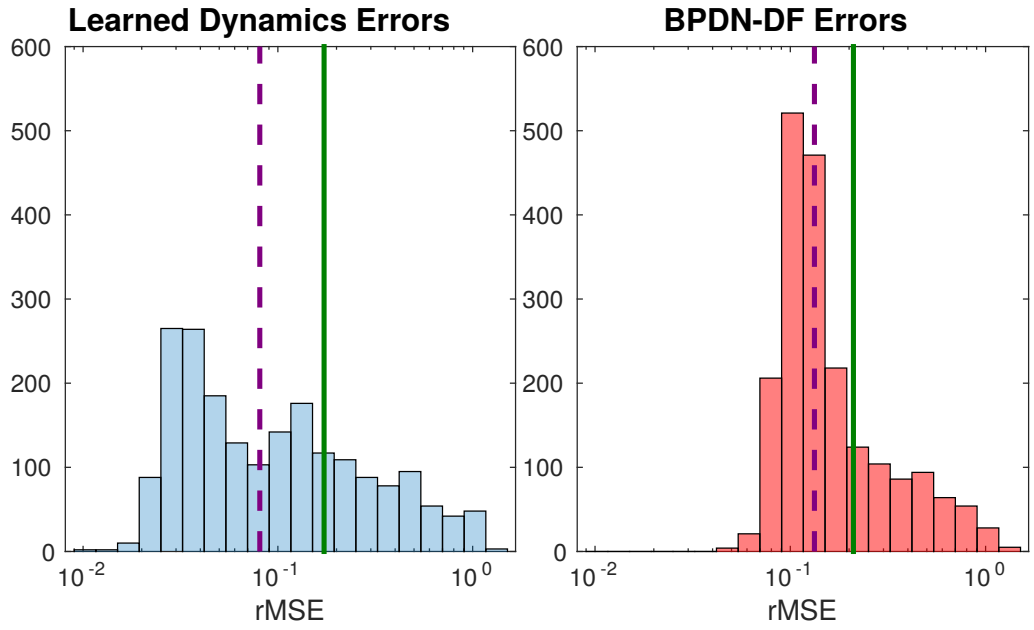

Figure 27: The proposed model using learned dictionaries recovers video patches with an average of 33%lower rMSE. While BPDN-DF has most errors cluttered around the 13% error area, our model reduces those errors to less than 8%.

CHAPTER V

SPATIALLY CORRELATED INFERENCE IN HYPERSPECTRAL IMAGERY[1]

5.1 Hyperspectral Imagery

Hyperspectral imagery (HSI) is a spectral imaging modality that obtains environmental and geographical information by imaging ground locations from airborne or spaceborne platforms. While multispectral imagery (MSI) acquires data over just a few (e.g., 3-10) irregularly spaced spectral bands, HSI typically uses hundreds of contiguous bands that are regularly spaced from infrared to ultraviolet. For example, the Worldview II MSI satellite [134] uses eight bands to represent the wavelengths from 0.435μm to 1.328μm, while typical HSI has approximately 60 bands over the same range in addition to many more bands at higher wavelengths. With spatial resolutions as low as 1m, the increased spectral resolution of HSI means that estimated ground reflectance data can be used to determine properties of the scene, including material classification, geologic feature identification, and environmental monitoring. A good overview of HSI and the associated sensors can be found in [135].

Exploiting HSI is often difficult due to the particular challenges of the remote sensing environment. For example, even "pure" pixels composed of a single material would have reflectance spectra that lie along a nonlinear manifold due to variations in illumination, view angle, material heterogeneity, scattering from the local scene geometry, and the presence of moisture [135, 136]. Additionally, pure pixels are essentially impossible to actually observe due to material mixtures within a pixel and scattering from adjacent areas [135]. One of the most common approaches to determining the material present in a given pixel $\boldsymbol{x}$

[1]This chapter is in collaboration with Dr. Bruno Olshausen (Sections 5.3) and Dr. Nicholas Tufillaro (Section 5.5). Further details about the work presented in this chapter is available in [20–25].

(called "spectral unmixing" [137]) is to use a linear mixture model such as in Equation (4), where $\{\boldsymbol{\psi}_k\}$ is a dictionary of approximation elements, $\{a_k\}$ are the decomposition coefficients and $\boldsymbol{\epsilon}$ is additive noise. Note that $\{\boldsymbol{x}, \boldsymbol{\psi}_k, \boldsymbol{\epsilon}\} \in \mathbb{R}^N$, where N is the number of spectral bands and the vectors are indexed by λ (which is suppressed in our notation). When the dictionary represents spectral signatures of the various material components present in the scene, they are typically called "endmembers" and the resulting coefficients (assumed to sum to one) represent the material abundances in each pixel. The endmember vectors are conceptualized as forming a convex hull about the HSI data (e.g., see the red vectors in Figure 28). Such a decomposition is often used for detecting the presence of a material in the scene or classifying the materials present in a pixel. A number of methods have been proposed for determining endmembers, including algorithms which select endmembers from the data based on a measure of pixel purity [138] or the quality of the resulting convex cone [139], tools that assist in the manual selection of endmembers from the data [140], algorithms which optimize endmembers for linear filtering [141], methods based on finding convex cones using principal component analysis (PCA) or independent component analysis (ICA) decompositions [142–145], iterative statistical methods that optimize the resulting convex cone [146], and iterative measures to select optimal endmember sets from larger potential sets [147]. However, these algorithms either rely on postulating candidate endmember sets for initialization [147], assume the existence of pure pixels in the scene [138, 139], attempt to encompass the data within a cone rather than directly represent the data variations [140, 142, 146], use orthogonal linear filters to attempt to separate out highly non-orthogonal spectra [141], or attempt to determine spectral statistics from decompositions in the spatial dimensions rather than the spectral dimension. [144, 145]. None of these methods attempt to directly *learn* from the spectral data a good representation of the low-dimensional, non-linear spectral variations inherent in HSI.

In addition to the difficulties determining the basic spectral components of an HSI

dataset, there are many resource costs (i.e., time, money, computation, availability of sensor platforms) that result from the high dimensionally of the data. During data acquisition, the high resolution of HSI data comes at the expense of sophisticated sensors that are costly and require relatively long scan times to get usable SNRs. After data acquisition, it is evident that reducing the dimensionality while retaining the exploitation value of the data would save significant computational and storage resources. If the higher-order statistics of the HSI data can be characterized, this information can be used to perform both dimensionality reduction of existing high-dimensional data and high-resolution inference from low-resolution data (collected from either a cheaper MSI sensor or a modified HSI sensor measuring coarse spectral resolution, thereby lowering scan times). One common approach to dimensionality reduction is PCA. However, the underlying Gaussian model in PCA means that it can only capture pairwise correlations in the data and not the higher-order (and non-Gaussian) statistics present in HSI data.

Following on developments in the computational neuroscience community, the signal processing community has recently employed signal models based on the notion of *sparsity* to characterize high-order statistical dependencies in data and yield state-of-the-art results in many signal and image processing algorithms [148]. Specifically, this approach models a noisy measurement vector $\boldsymbol{x}$ as being generated by a linear combination of just a few elements from the dictionary $\{\boldsymbol{\psi}_k\}$. This is the same model as in (4), but where the coefficients are calculated to have as few non-zero elements as possible. Much like PCA, sparse coding can be viewed as a type of dimensionality reduction where a high dimensional dataset is expressed in a lower dimensional space of active coefficients. However, while PCA calculates just a few principal components and uses essentially all of them to represent each pixel, sparse coding models typically employ a larger dictionary but use only a few of these elements to represent each pixel. When cast in terms of a probabilistic model, this sparsity constraint corresponds to a non-Gaussian prior that enables the model to capture higher order statistics in the data.

Due to the high spatial resolution of modern HSI sensors (resulting in just a few dominant materials in a pixel), sparsity models seem especially relevant for this sensing modality. In fact, initial research into sparsity models for spectral unmixing in HSI has shown promising results [149,150]. While a sparse decomposition can be estimated for any dictionary, previous research [32] has shown that unsupervised learning techniques can be used in conjunction with an example dataset to iteratively learn a dictionary that admits optimally sparse coefficients (without requiring the dataset to contain any "pure" signals that correspond to a single dictionary element). These methods leverage the specific high-order statistics of the example dataset to find the underlying low-dimensional structure that is most efficient at representing the data.

In contrast to the typical endmember model described above, the sparse coding model does not assume that the data lie within the convex hull of the dictionary. Instead, the learned sparse coding dictionary elements will tend to look like the basic spectral signatures comprising the scene (early encouraging evidence of this can be found in [151]). In fact, the sparse coding model may actually learn several dictionary elements to represent some types of materials, especially when that material spectra demonstrates highly nonlinear variations within the scene. Because of the sparsity constraint, one would expect these learned dictionaries to reflect the specific statistics of the HSI data by locally approximating these nonlinear data manifolds [152] (as illustrated in Figure 28, and in contrast to typical endmember models that form a convex hull containing the data).

We have modified the unsupervised learning approach described in [32] (or see Section 2.3) and applied it to HSI data to learn a dictionary that is optimized for sparse coding. Importantly, the HSI dataset used in this study has significant ground truth labeling of material classes making it possible to examine the characteristics of the learned dictionary relative to the data. Using this learned dictionary, we make three main contributions. First, we show that the sparse coding model learns meaningful dictionaries that correspond to

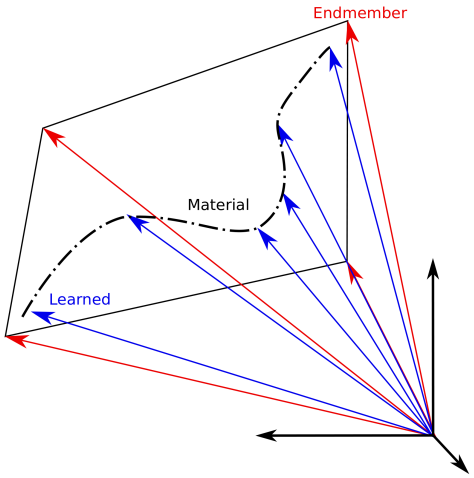

Figure 28: Typical endmember analysis uses vectors that compose a convex hull around the data. In this stylized illustration, the data manifold is indicated by the dashed line and the red vectors represent the endmembers. In contrast, a learned dictionary for sparse coding attempts to learn a local approximation of the nonlinear data characteristics directly (indicated here by blue vectors).

known spectral signatures: they locally approximate nonlinear data manifolds for individual materials, and they convey information about environmental properties such as moisture content in a region. Second, we generate simulated imagery at MSI-level resolution and show that the learned HSI dictionaries and sparse coding model can be effectively used to infer HSI-resolution data with very high accuracy (even for data of the same region collected in a different season). Finally, we use ground truth labels for the HSI data to demonstrate that a sparse coding representation improves the performance of a supervised classification algorithm, both in terms of the classifier complexity (i.e., classification time) and the ability of the classifier to generalize from very small training sets.

5.2 Background and Related Work

5.2.1 Methods

Given a pixel $\boldsymbol{x} \in \mathbb{R}^N$ and a fixed dictionary $\{\boldsymbol{\psi}_k\}$ with $\boldsymbol{\psi_k} \in \mathbb{R}^N$ for $k \in [1, \ldots, M]$, the goal of sparse coding is to find a set of coefficients that represent the data well using as few non-zero elements as possible. Written mathematically, the goal is to minimize an objective function that combines data fidelity and a sparsity-inducing penalty. A common choice is to use a regularized least-squares objective function such as Equation 3 [153]. This objective is convex in the coefficients when the dictionary is fixed, meaning that solving $\{a_k\} = \arg\min_{\{a_k\}} J_\gamma(\boldsymbol{x}, \{a_k\}, \{\boldsymbol{\psi}_k\})$ is a tractable optimization. This general approach is applicable directly to HSI with one small modification: we constrain the coefficients to be non-negative ($a_k \geq 0$) to maintain physical correspondence between the coefficients and the relative abundance of material spectra present in the scene. Due to its wide use in the community, its ability to enforce positive coefficients without a sum-to-one constraint, and established reputation for quick convergence, we use the specialized optimization package described in [34] to solve this constrained optimization and calculate sparse coefficients. While other solvers have been explored in the specific context of HSI [154, 155] that may be faster in some settings, many of these HSI-specific solvers include additional constraints which we do not employ (e.g. $\sum_k |a_k| = 1$). The framework we present here is largely agnostic to the specific solver as long as it returns accurate solutions, so other choices could be substituted if there were advantages for a given application. A detailed analysis of various algorithms to optimize (3) in the context of HSI unmixing is given in [150].

An alternate interpretation of the cost function in (3) is to consider the problem as Bayesian inference as in Section 2.1. This Bayesian formulation allows us to naturally extend the sparse approximation problem to more general observation models and inverse problems, such as the high resolution inference task described in Section 5.3.2 and similar inverse problems described in [156]. Note also that the sparse prior introduces a non-Gaussianity into the model that is critical for capturing the high-order data statistics. An

approach such as PCA that fundamentally assumes a Gaussian data model can only learn from pairwise correlations in the data, and is therefore unable to capture the higher-order statistics.

To learn an optimal dictionary for sparse coding, we follow the same basic outline as in Section 2.3 and [32, 57]. As with the coefficient optimization, other algorithms have been proposed for the learning step that could be substituted for this steepest decent approach. In particular, many other methods (including the recently proposed K-SVD) use second order information in the learning step to reduce the number of learning iterations required for convergence (though this may come at the cost of increasing the batch size per iteration to get better estimates for the update step) [33, 157].

The results in [32, 57] demonstrate that this unsupervised approach can start with an unstructured random dictionary and recover known sparse structure in simulated datasets, as well as uncover unknown sparse structure in complex signal families such as natural images. We again adopt this general approach with a small modification: we constrain the dictionary elements to be non-negative ($\boldsymbol{\psi}_k \geq 0$) to maintain physical correspondence with spectral reflectances. To be concrete, the dictionary learning method we use is specified in Algorithm 3, and we determine convergence visually by when the dictionary elements stopped adapting. In our experience, most of the dictionary elements were well-converged by 1000 iterations of the learning step (approximately 50 minutes of computation on an 8-core Intel Xeon E5420 with 14GB of DDR3 RAM). Some dictionary elements corresponding to less prominent materials (that are randomly selected less often during learning) seem to require 10,000-20,000 learning iterations to converge (approximately 10-15 hours on the same machine). We often conservatively let the algorithm run for 20,000 to 80,000 iterations at a smaller step size to assure good convergence. The increasing prevalence of parallel architectures in multi-core CPUs and graphics processing units should provide increasing opportunities to speed up this type of unsupervised learning approach.

Algorithm 3 Sparse coding dictionary learning algorithm of [57], modified for HSI.

Set $\gamma = 0.01$
Set $\mu = 10$
Initialize $\{\boldsymbol{\psi}_k\}$ to random positive values
repeat
  for $i = 1$ to 200 **do**
    Choose HSI pixel $\boldsymbol{x}$ uniformly at random
    $\{a_k\} = \arg\min J(\{a_k\}, \{\boldsymbol{\psi}_k\})$ s.t. $a_k \geq 0$
    $\Delta\boldsymbol{\psi}_l(i) = a_l\left(\boldsymbol{x} - \sum_{k=1}^{M} \boldsymbol{\psi}_k a_k\right)$
  end for
  $\boldsymbol{\psi}_l \leftarrow \left[\boldsymbol{\psi}_l + \frac{\mu}{200}\sum_i \Delta\boldsymbol{\psi}_l(i)\right]_+$
  $\mu \leftarrow 0.995\mu$
until $\{\boldsymbol{\psi}_k\}$ converges

Finally, we note that the proposed approach can have local minima or non-unique solutions in at least two respects, especially in the case of HSI. First, though the coefficient optimization using an ℓ_1 sparsity penalty is convex, the ideal ℓ_0 sparse solution may not be unique when the one-sided coherence of the dictionary $\max_{i\neq j} |\langle\boldsymbol{\psi}_i, \boldsymbol{\psi}_j\rangle|/\|\boldsymbol{\psi}_i\|_2^2$ is large [158]. Second, though there are few analytic guarantees about the performance of dictionary learning algorithms, recent results indicate that the ideal dictionary is more likely to be a local solution to the optimization presented here when the coherence of the dictionary is also low [159]. Since many materials have spectral signatures with high correlation in some bands, typical HSI dictionary databases have coherence values very close to unity [150], and we observe similar values in our learned dictionaries. Despite not being favorable for the technical results described above regarding coefficient inference and dictionary learning, the inferred coefficients and learned dictionaries appear to be robust and useful in the applications described here. Indeed, it is likely in these cases that despite there being many local solutions (and a unique minima perhaps even not existing), many of the suboptimal solutions are also quite good and useful in applications. In particular, we have repeated the dictionary learning experiments described in this paper many times (with different random initial conditions), with no significant changes in the qualitative nature of the dictionary or the performance in the tasks highlighted in Section 5.3. This also corresponds to the

results in [150] showing that despite the near-unity coherence in a standard hyperspectral endmember dictionary, these dictionaries can yield good sparse representations useful in spectral unmixing applications.

5.2.2 Hyperspectral dataset and learned dictionaries

In this paper we apply the dictionary learning method described in Algorithm 3 to learn a 44-element dictionary for a HSI scene of Smith Island, VA. This scene has 113 usable spectral bands (ranging from 0.44–2.486μm) acquired by the PROBE2 sensor on October 18, 2001.[2] The data has a spatial resolution of approximately 4.5m and was postprocessed to estimate the ground reflectance. Of the 490,000 pixels in the dataset, 2700 pixels are tagged with ground truth labels drawn from 22 categories. These categories include specific plant species and vegetation communities common to wetlands, and were determined by *in situ* observations made with differential GPS aided field studies during October 8–12, 2001. More information about the HSI dataset and the ground truth labels can be found in [160–163]. The size of the dictionary (44 elements) was made to ensure that there were multiple elements available for each of the 22 known material classes in this particular dataset. The number 44 represented a compromise between smaller dictionaries that didn't perform as well on the tasks described in Section 5.3 (especially the local manifold approximation), and larger dictionaries that presented more difficulty getting all of the elements to converge in the learning.[3] In general, determining the optimal number of dictionary elements to learn for a dataset is an open question and could be a valuable future research direction.

We cross-validated the results of this paper in two ways. First, 10,000 randomly selected pixels were excluded from the dataset before the dictionary learning so that they

[2] Smith Island is a barrier island that is part of the Virginia Coast Reserve Long-Term Ecological Research Project. For more details, see http://www.vcrlter.virginia.edu. This dataset was generously provided by Charles Bachmann at the Naval Research Laboratory.

[3] While performance in the signal processing tasks we tested did improve with larger dictionaries, we note that the performance difference was often relatively minor when using 22 element dictionaries and this size would likely be sufficiently for this dataset in many applications.

could be used in testing. Second, we also have available data from another HSI collection of the same geographic region using the same sensor on August 22, 2001. While this is close enough in time to assume that there are no major geologic changes in the scene, this data does come from a different season where the vegetation and atmospheric characteristics are potentially different, resulting in different statistics from the data used in the learning process. We use this dataset specifically to assess the potential negative effects of mismatch between the statistics of the training and testing datasets when performing signal processing applications using the learned dictionary.

5.2.3 Related work

Prior work in using unsupervised methods to learn HSI material spectra has used some algorithms that are very related to our present approach. For example, ICA can be viewed as finding linear filters that give high sparsity, and prior work [143,164,165] demonstrates that ICA can be effective at determining a range of spectral signatures from preprocessed data. Other approaches also based on Bayesian inference (but not necessarily a sparsity-inducing prior) [166] have been used to learn HSI dictionaries, but this approach has trouble including information from large datasets and often uses ICA as a preprocessing stage to reduce the number of pixels to analyze. The technique most closely related to our current approach is blind source separation based on non-negative matrix factorization (NMF) [167, 168]. While not explicitly incorporating sparsity constraints, results using NMF have been shown to exhibit sparse behavior [169]. In the NMF setup, the sparsity level of the decomposition is difficult to control [169] and previous work in [167] mitigates this by adding an explicit sparsity inducing term. Additionally, the above mentioned approaches all retain the sum-to-one constraint, which we drop due to the variable power in the pixels throughout the scene.

In addition to these results on unsupervised learning, as well as additional encouraging prior work on using sparsity models for spectral unmixing [149, 150] and learning dictionaries that resemble material spectra [151], Castrodad et al. [170] have explored using a

sparsity model and learned dictionaries to improve supervised classification performance on HSI data.[4] In Section 5.3.3 we will explore the advantages of using sparse coefficients from a learned dictionary in an off-the-shelf classification algorithm. In [170], the authors use labeled data to learn a separate dictionary for each class and classify data by determining which of these candidate dictionaries best describes an unknown pixel (defined by having the minimum value for the objective function in equation (3)). This approach is customized to the classification problem, and we expect the classification performance would outperform the general approach we describe in Section 5.3.3. In contrast, the approach in [170] requires a more computationally expensive learning process (due to the multiple dictionaries), requires labeled data before the learning process, and generates a dictionary that is tailored to the classification task and may not generalize as well to other tasks.

Zhou et al. [156] have explored using a sparsity model and learned dictionaries to effectively solve inverse problems in HSI. In Section 5.3.2 we will explore the ability of sparse coefficients from a learned dictionary to infer high resolution spectral data from low resolution imagery by formulating the task as a linear inverse problem. In [156], the authors show that when removing substantial amounts of data from an HSI datacube, a learned dictionary can be used to exploit the correlation structure present in each band to infer the missing data and reconstruct the spatial image associated with each band. This inpainting task is a very similar inverse problem to the one we examine in Section 5.3.2, differing primarily in the type of measurement operator used in the model (i.e., blurring vs. subsampling) and the dimension of the data used in the learning and reconstruction (i.e., spectral vs. spatial).

5.3 Analyzing the Learned Dictionary

5.3.1 Learned Dictionary Functions

While the learning procedure described in Algorithm 3 adapts the dictionary to the high-order statistics of the HSI data, there are no constraints added that ensure the resulting

[4]The authors in [170] use a different learning algorithm (K-SVD [33]) from our gradient approach, but it is attempting to achieve the same goal of learning an optimal dictionary for sparse approximation.

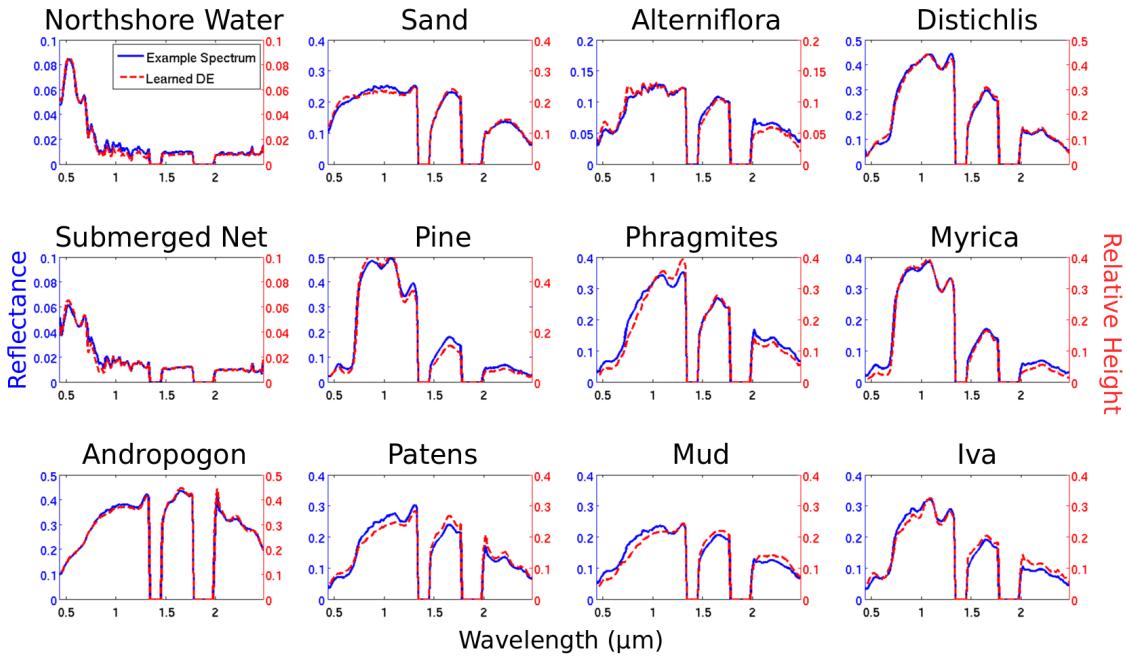

Figure 29: Example spectra for materials in the labeled classes of the Smith Island dataset and the learned dictionary element (DE) that is the closest match for each example. The two obvious gaps in the spectra are bands removed from consideration in the original dataset due to the interactions with the atmosphere in these regions.

dictionary elements will correspond to physical spectra or be informative about material properties in the scene. To examine the properties of the learned dictionary, examples elements are plotted in Figure 29. It is clear that these dictionary elements not only have the general appearance of spectral reflectances, they also match the spectral signatures of many of the materials that are known to be in the scene. Using the ground truth labels from the Smith Island dataset (which denote the dominant material present in the pixel), Figure 29 shows an example spectral signature from a class along with the dictionary element that has the largest coefficient in the sparse decomposition of that pixel. Despite being given no *a priori* information about the data beyond the sparsity model (i.e., without being given the class labels and corresponding pixels), the algorithm learns spectral shapes that correspond to a number of component material spectra present in the image. These learned dictionaries cover a wide variety of distinct material classes for which we have ground truth labels,

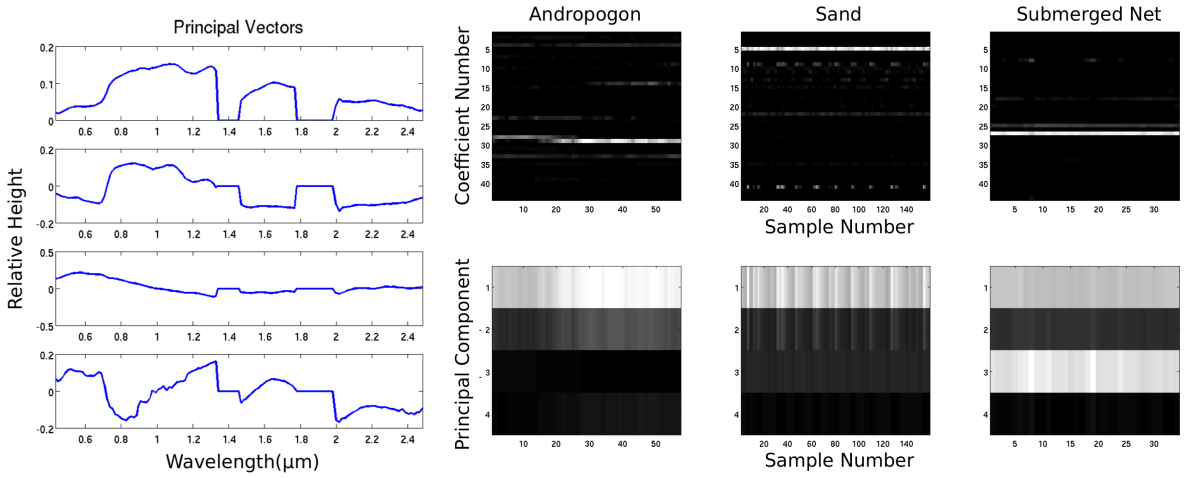

Figure 30: (Left) The top four principal components for the Smith Island dataset (capturing 99.9% of the variance). In contrast to the learned dictionary elements in Figure 29, only one of the principle components looks generally like a spectral signature. (Right) PCA and sparse coding coefficients representing every sample of the data from three of the labeled classes ("Andropogon", "Sand", and "Submerged Net"). The brightness at each pixel represents the intensity of a given coefficient for a specific pixel. Note that PCA uses many of the same coefficients for different materials (e.g., coefficient 1 is always used), while sparse coding tends to select distinct coefficients for the different materials.

including "Pine", "Water", "Mud" and "Distichlis", as well as very similar spectra, such as "Water" and "Submerged Net" or "Pine trees" and "Iva".

In contrast, Figure 30 shows the first four principal components found through PCA analysis on the same HSI dataset, which is sufficient to capture 99.9% of the variance in the data. While the first principal component does have some similarity to a general vegetation spectrum, the other spectral components do not correspond to physically meaningful spectral features. Figure 30 also shows the comparison between the decomposition coefficient in the sparsity model and PCA for all pixels in four of the labeled classes. The raster plots show that while the sparse decomposition and the principal components both only need a few coefficients to represent the data, the sparse decomposition chooses different coefficients for different spectral shapes (i.e., the material information is encoded in the selection of active coefficients) whereas PCA uses the same four vectors to represent nearly

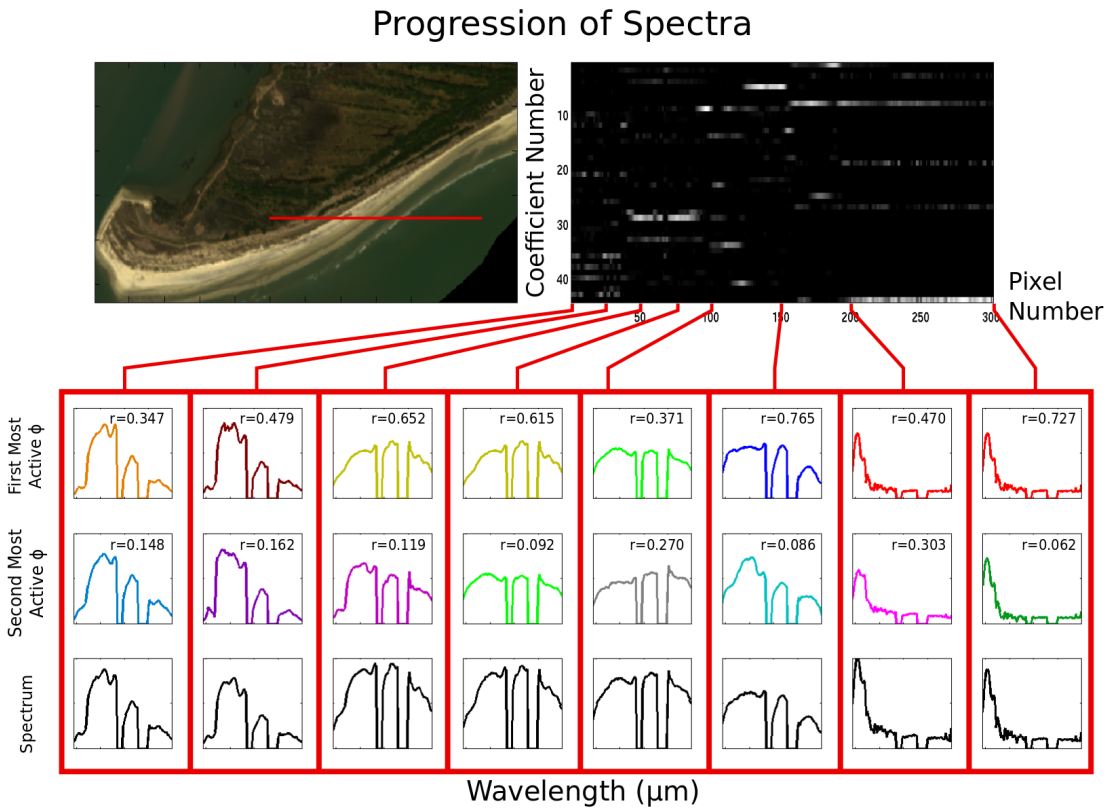

Figure 31: Progression of sparse coding coefficients from a row of contiguous pixels in the Smith Island dataset. (Upper left) The red line indicates a row of 300 pixels selected for analysis. These pixels (numbered left to right) represent a progression from an inland region to the water off the east coast of the island. (Upper right) The sparse coding coefficients for the row of pixels is shown, where the brightness of a pixel indicates the intensity of each coefficient for each pixel. Note that many of the same coefficients are often active in the same geographic regions, and the progression from one type of element to another (e.g., sand to water) can be seen by different coefficients dominating the decomposition. (Bottom) The spectra for pixels 1, 25, 50, 75, 100, 150, 200 and 300 are shown in the bottom row (in black), along with the two most active dictionary elements in the top two rows (color coded). The fractional abundance for each dictionary element in each pixel is given by $r = |a_i|/||\mathbf{a}||_1$. Note that many of the same dictionary elements can be seen dominating the decomposition in regions with similar material composition.

all of the data. This comparison illustrates that the learned dictionary under the sparsity model has a much closer correspondence to the individual spectral characteristics found in the dataset than PCA, indicating that this representation may have many advantages for tasks such as classification.

While it is clear that the dictionary elements are learning spectral elements present in the scene, this representation will be most meaningful if there is consistency in the way environmental features are represented. In other words, when looking across the scene, do the sparse decompositions change in a way that reflects the changes in the underlying geologic features? We extracted a row of pixels from the Smith Island dataset, starting inland and ending in the water off the coast of the island. The selected row of pixels is shown in red in Figure 31, superimposed on a magnified RGB rendering of that portion of the island. Figure 31 shows the coefficient decomposition of each pixel, as well as the measured spectrum and the two most active dictionary elements at various locations along the row. Included with each of the two most active dictionary elements is the fractional abundance $r = |a_i|/\|\mathbf{a}\|_1$ of that dictionary element in the decomposition. This row starts with mostly vegetation spectra for the first 75 pixels, changing to sand-like spectra by the the 100^{th} pixel and eventually to water spectra by the 160^{th} pixel.

We highlight two important properties of the coefficient decompositions over the pixel progression in the raster plot in Figure 31. First, the sparse coefficients are relatively consistent over contiguous spatial ranges, with the same small sets of coefficients generally dominating the decomposition over small contiguous regions. While this is evident in the regions dominated by sand and water, there are also repeated dictionary elements across several spatial locations in the regions dominated by vegetation (which we would expect to have much more variability over pixels with 4m resolution). Second, some slowly changing geologic properties are actually observable in the gradual onset and offset of specific dictionary elements in the decomposition. One prominent example of this is the slow change from dictionary element 8 to dictionary element 44 over the span of water moving away from the shoreline, indicating the slow fading of shallow water to deep water (which have different spectral characteristics and are represented by different dictionary elements). Another example of this is the rise of dictionary element 9 from the second most active to the most active element from pixels 75 and 100, indicating the slowly increasing presence of a

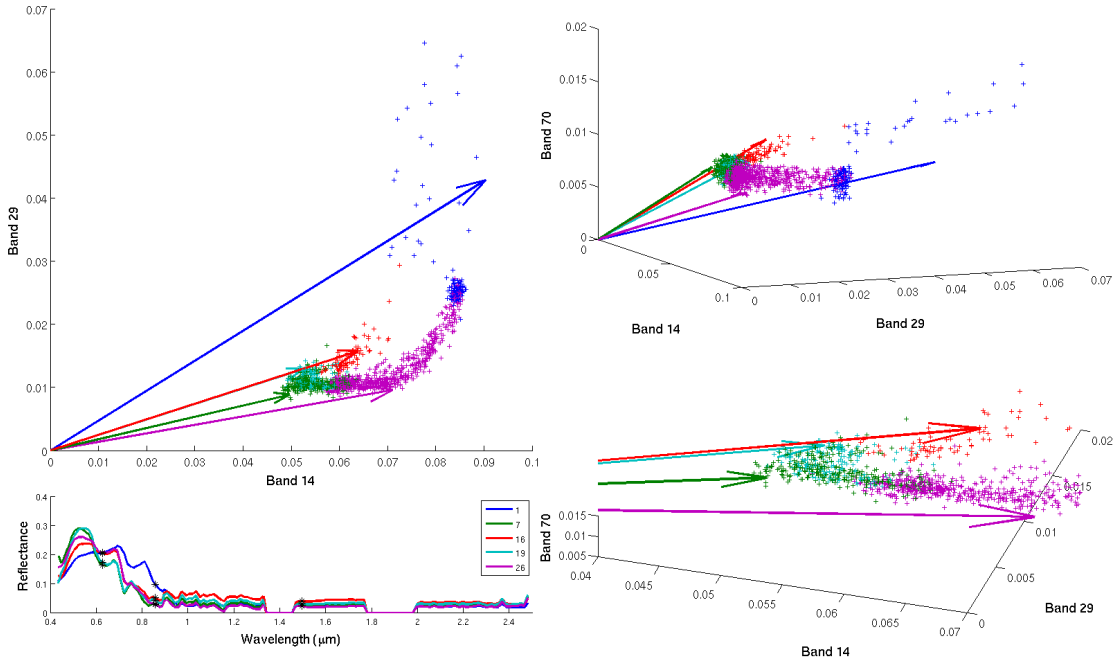

Figure 32: The nonlinear structure of water pixels is locally approximated by the learned dictionary. The plots in the upper left, upper right and bottom right all show the spectra water pixels (selected from a contiguous region) projected onto three spectral bands (14,29,70). Even in three dimensions, it is clear that the data live on a nonlinear manifold, and there is clear structure in the variability. The vectors represent the projection onto the same three bands of five learned dictionary elements. The points representing water pixels are color coded to indicate which dictionary element has the largest value when inferring the sparse coefficients, showing that contiguous values on the manifold are coded using the same dictionary element.

particular vegetation characteristic in this region.

In addition to the spectral matches shown in Figure 29 and the spatial coefficient variations shown in Figure 31, another important aspect of the learned dictionary is to examine how it represents the nonlinear variations within a particular material class [136].

For example, Figure 32 shows full spectral signatures for a patch of water off the coast of Smith Island, as well as spectral bands 14, 29 and 70 (0.6278, 0.8572 and 1.4962 μm) from three different view angles to show the geometry of these points in 3-D spectral space.[5] Despite being one material class ("water"), it is evident even in these few bands that the

[5]These are the same spectral bands and approximately the same region highlighted in Figure 1 of [136].

measured spectrum lies on a nonlinear manifold. Superimposed on the 3-D spectral plots are five of the learned dictionary elements projected onto these same three spectral bands. The measured spectra are color coded to indicate which of these five learned dictionary elements are dominant in their sparse decomposition. The contiguity of this color coding over small manifold regions demonstrates that rather than containing the measured spectra in a convex hull, the learned dictionaries are essentially forming a local linear approximation to this manifold. So, despite being a linear data model, the dictionary learns multiple elements that capture the nonlinear spectral variations by locally approximating the manifold structure in a meaningful way. In our experiments with other endmember extraction algorithms such as [139], the learned sparse dictionary does appear to produce a representation that more closely tracks the nonlinear variations in the data points (e.g., produces a smaller relative MSE between the data and the dictionary elements) compared to a method restricted to finding a convex cone around the data. A more detailed characterization of the differences between various linear models at representing nonlinear material variations would be a valuable direction for future research.

5.3.2 Reconstructing HSI-resolution from MSI-resolution data

As discussed earlier, while the high spectral resolution of HSI is valuable, acquiring data at this resolution comes at a cost. In terrestrial remote sensing, hyperspectral imagers are relatively rare instruments, and it would be much more resource efficient to perform most spectral imaging at MSI-level resolution. Data at this resolution could either be gathered by actual MSI sensors, or by HSI sensors modified to decrease their spectral resolution (which could potentially decrease scan times). The question we consider here is whether a dictionary learned on an HSI training set could be used to accurately infer high resolution spectra from subsequent data collected at MSI-level spectral resolution.

In this basic paradigm, assume that we start with a learned dictionary that has been adapted to the specific structure of the desired HSI data. This could arise from earlier HSI of the scene being imaged, or imaging from other geographic regions with similar

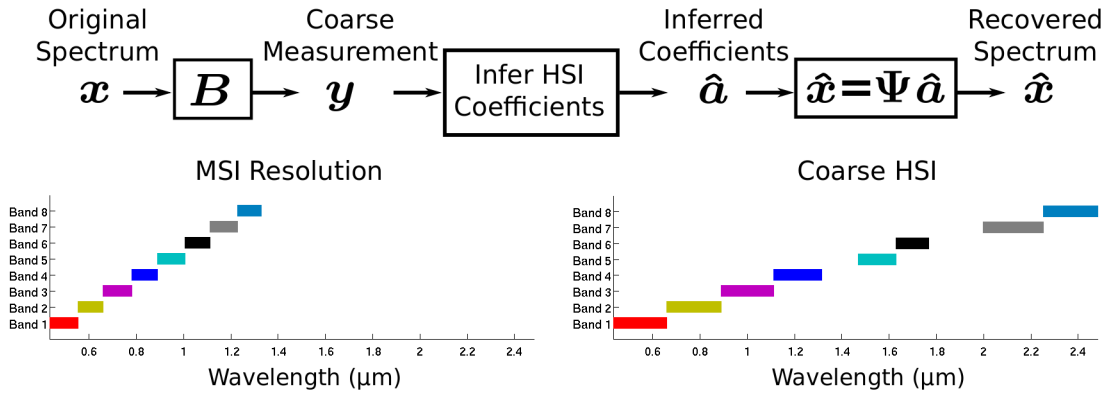

Figure 33: Reconstructing spectra with HSI resolution from measurements with MSI-level resolution. (Top) A schematic of the process for simulating low resolution spectral data and performing recovery. The matrix $\boldsymbol{B}$ characterizes the measurement process (i.e., the sensitivity function of the sensor), simulating the aggregation of high resolution spectral information into low resolution spectral bands. (Bottom left) A diagram indicating the sensitivity function for MSI resolution measurements, where 113 HSI bands are collapsed into 8 equally spaced measurement bands over the lowest wavelengths (approximately matching the spectral bands reported by the Worldview II MSI sensor). Note that no information is measured from the highest wavelength regions. (Bottom right) A diagram indicating the sensitivity function for coarse HSI measurements, where 113 HSI bands are collapsed into 8 nearly equally spaced measurement bands across the whole HSI spectrum.

environmental features (and therefore similar statistics). For the new data acquired at MSI-level resolution, we assume for a first approximation that each band is a linear combination of some group of spectral bands in the underlying true HSI data. Specifically, we model the MSI-resolution data as

$$\boldsymbol{y} = \boldsymbol{B}\boldsymbol{x} + \epsilon = \boldsymbol{B}\sum_{k=1}^{M} \boldsymbol{\psi}_k a_k + \epsilon, \tag{66}$$

where $\boldsymbol{y} \in \mathbb{R}^L$ ($L < N$) is the new coarse resolution data and $\boldsymbol{B}$ can be thought of as an ($L \times N$) "blurring" matrix that bins the spectral bands of the desired HSI data. While $\boldsymbol{B}$ could be any matrix describing the sensitivity function of the imager acquiring the MSI-resolution data, we will consider $\boldsymbol{B}$ that simply sums spectral bands over a contiguous range.

This measurement paradigm fits nicely into the well-known framework of Bayesian inference (or equivalently, linear inverse problems in image processing). Essentially, given the wealth of information about the statistics of the HSI we would like to obtain, Bayesian

Table 2: Relative recovery error for HSI spectra from coarse HSI measurements (full spectrum). Results are reported for testing data collected on the same day (SD) as the training data used to learn the dictionary, as well as results for testing data collected on a different day (DD).

	Mean Error	Median Error
44 Learned DE (SD)	$8.249\text{x}10^{-4}$	$4.911\text{x}10^{-4}$
44 Learned DE (DD)	$7.054\text{x}10^{-3}$	$6.005\text{x}10^{-3}$
44 Exemplar DE (SD)	$6.280\text{x}10^{-3}$	$2.709\text{x}10^{-3}$
44 Exemplar DE (DD)	$1.493\text{x}10^{-2}$	$1.105\text{x}10^{-2}$
44 Random DE (SD)	$4.143\text{x}10^{-1}$	$4.524\text{x}10^{-1}$
44 Random DE (DD)	$3.965\text{x}10^{-1}$	$4.165\text{x}10^{-1}$

inference allows one to optimally answer the question of what underlying HSI data $\boldsymbol{x}$ is most likely given the observed MSI-resolution data $\boldsymbol{y}$. Specifically, given the new model in (66), the likelihood of the data $\boldsymbol{y}$ given the coefficients $\{a_k\}$ is now the Gaussian distribution

$$p(\boldsymbol{y}|\{a_k\}) \propto e^{-\frac{1}{2\sigma_\epsilon^2}} \left\| \boldsymbol{y} - \boldsymbol{B} \sum_k \boldsymbol{\psi}_k a_k \right\|_2^2 .$$

We can again use an independent Laplacian prior on the sparse coefficients $\{a_k\}$, and write the posterior distribution using exactly the same simplifications as before. The optimal MAP estimate of the sparse coefficients given the observed data $\boldsymbol{y}$ is therefore given by optimizing the following objective function with respect to the coefficients:

$$\widetilde{J}_\gamma(\boldsymbol{y}, \{a_k\}, \{\boldsymbol{\psi}_k\}) = \left\| \boldsymbol{y} - \boldsymbol{B} \sum_{k=1}^{M} \boldsymbol{\psi}_k a_k \right\|_2^2 + \gamma \sum_k |a_k|. \tag{67}$$

This optimization program is very similar to (3) (and can be solved by the same software packages), but incorporates the measurement process described by $\boldsymbol{B}$ into the inference. Given the estimated sparse coefficients, the HSI vector $\boldsymbol{x}$ is reconstructed according to (4): $\widehat{\boldsymbol{x}} = \sum_k \boldsymbol{\psi}_k \widehat{a}_k$. The full workflow is shown schematically in Figure 33. We note that many linear inverse problems are formulated in a similar way depending on the choice of $\boldsymbol{B}$, including inpainting missing data such as the application considered by [156].

For proof-of-concept simulations we generated simulated data with MSI-level resolution from pixels that were not used in the training dataset, and perform the inference

Table 3: Relative recovery error for HSI spectra from MSI measurements (no measurements from highest wavelengths). Results are reported for testing data collected on the same day (SD) as the training data used to learn the dictionary, as well as results for testing data collected on a different day (DD).

	Mean Error	Median Error
44 Learned DE (SD)	$1.271\text{x}10^{-2}$	$1.791\text{x}10^{-3}$
44 Learned DE (DD)	$2.456\text{x}10^{-2}$	$1.219\text{x}10^{-2}$
44 Exemplar DE (SD)	$1.132\text{x}10^{-2}$	$5.552\text{x}10^{-3}$
44 Exemplar DE (DD)	$2.225\text{x}10^{-2}$	$2.135\text{x}10^{-2}$
44 Random DE (SD)	$7.845\text{x}10^{-1}$	$8.974\text{x}10^{-1}$
44 Random DE (DD)	$7.775\text{x}10^{-1}$	$9.946\text{x}10^{-1}$

process described above to estimate the high-resolution spectra from the low-resolution measurements. In the first set of simulations, the matrix $\boldsymbol{B}$ (illustrated in Figure 33) generates simulated data with 8 equally spaced bands covering the entire spectral range of the HSI data. This $\boldsymbol{B}$ is intended to model a hyperspectral imager collecting spectral data with an order of magnitude less spectral resolution than the original data. We used two testing datasets in this simulation: the 10,000 pixels from the October 2001 scan of Smith Island that were withheld from the learning process, and 10,000 randomly selected pixels from the August 2001 scan of the same geographic region. By using HSI collected on a different date we can examine the effects of using a dictionary that was learned on data with different statistics than the data we are trying to reconstruct (due to different vegetation characteristics in the different seasons and different atmospheric conditions present on the different days).

We infer the sparse coefficients in the HSI dictionary given the simulated MSI-resolution data by minimizing the objective function in Equation (67) as described above. For comparison purposes and to determine the value of the learning process in the reconstruction, we repeated this recovery process with a 44-element dictionary of random values (i.e., the initialization conditions for the dictionary learning) and with an exemplar dictionary formed by taking two random spectral signatures from each class in the original labeled HSI data (for a total of 44 dictionary elements). Figures 34 and 35 show examples of the

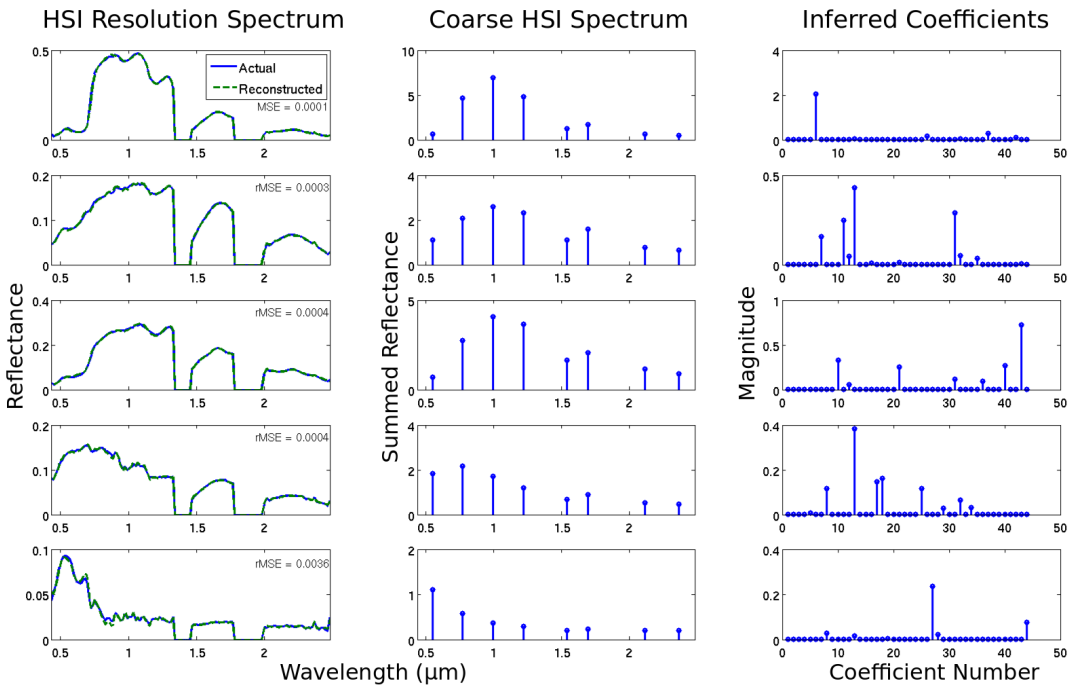

Figure 34: Reconstructing HSI data from simulated coarse HSI measurements using training and testing data collected on the same date. Plots show original HSI spectrum in blue (113 bands), simulated coarse HSI spectrum (8 bands), inferred sparse coefficients, and reconstructed HSI spectrum in green. Examples were selected to illustrate a range of recovery performance, from examples of the best recovery on top to examples of the worst recovery on the bottom.

original HSI, the simulated coarse resolution data, the estimated sparsity coefficients in the learned dictionary, and the subsequent recovered HSI data for the test datasets collected on the same date (SD) and a different date (DD) as the training dataset. The set of examples shown in the figure span the range of the most favorable and least favorable reconstructions.

Table 2 reports the average relative MSE for the reconstructions, calculated as

$$e_i = \frac{||\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i||_2^2}{||\boldsymbol{x}_i||_2^2}. \tag{68}$$

The aggregate results as well as the specific plotted examples demonstrate that the HSI-resolution data is recovered with less than 0.09% relative MSE for the SD testing set and less than 0.71% relative MSE on the DD testing set. While the reconstruction is worse on

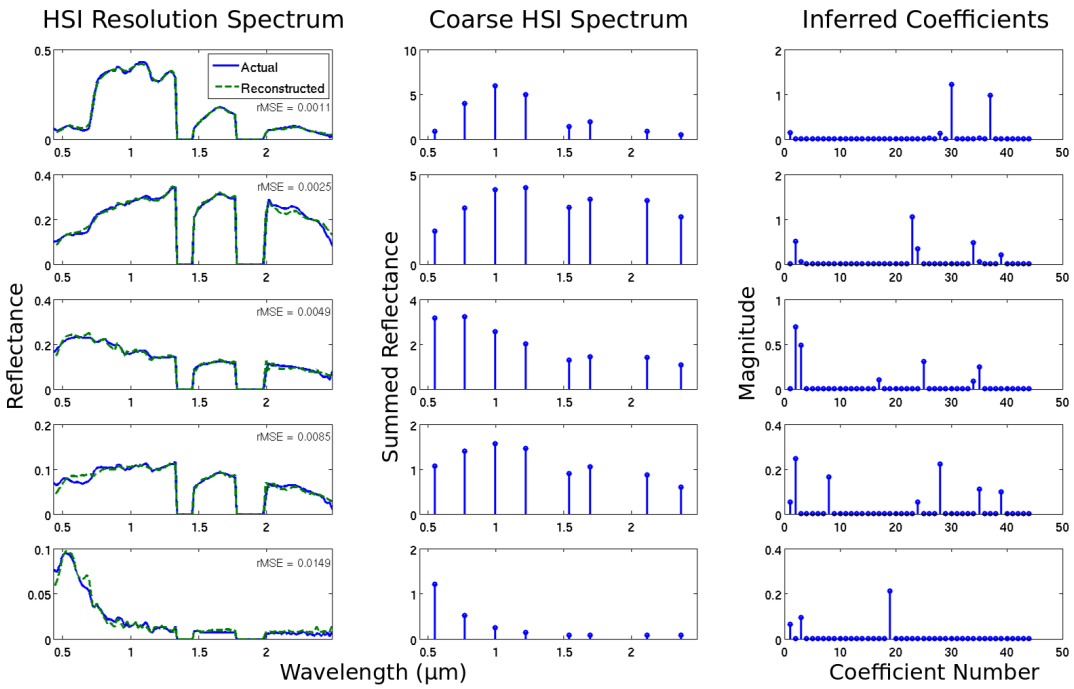

Figure 35: Reconstructing HSI data from simulated coarse HSI measurements using training and testing data collected on different dates (in different seasons). Plots show original HSI spectrum in blue (113 bands), simulated coarse HSI spectrum (8 bands), inferred sparse coefficients, and reconstructed HSI spectrum in green. Examples were selected to illustrate a range of recovery performance, from examples of the best recovery on top to examples of the worst recovery on the bottom.

the DD dataset because of the mismatch in the training and testing statistics, the reconstructions are still very good overall and often capture even fine detail in the HSI spectra. Also note that the learned dictionary is performing significantly better than both the exemplar dictionary (which was chosen using oracle knowledge of the classes to ensure good coverage of the various materials) and the random dictionary (indicating the value of the learning process).

In the second set of simulations, the matrix $\boldsymbol{B}$ (illustrated in Figure 33) generates simulated data with 8 equally spaced bands excluding the highest wavelength regions. This $\boldsymbol{B}$ is intended to model a multispectral imager, and we selected the bands to approximately match the reported bands of the WorldView II multispectral sensor. We used the same

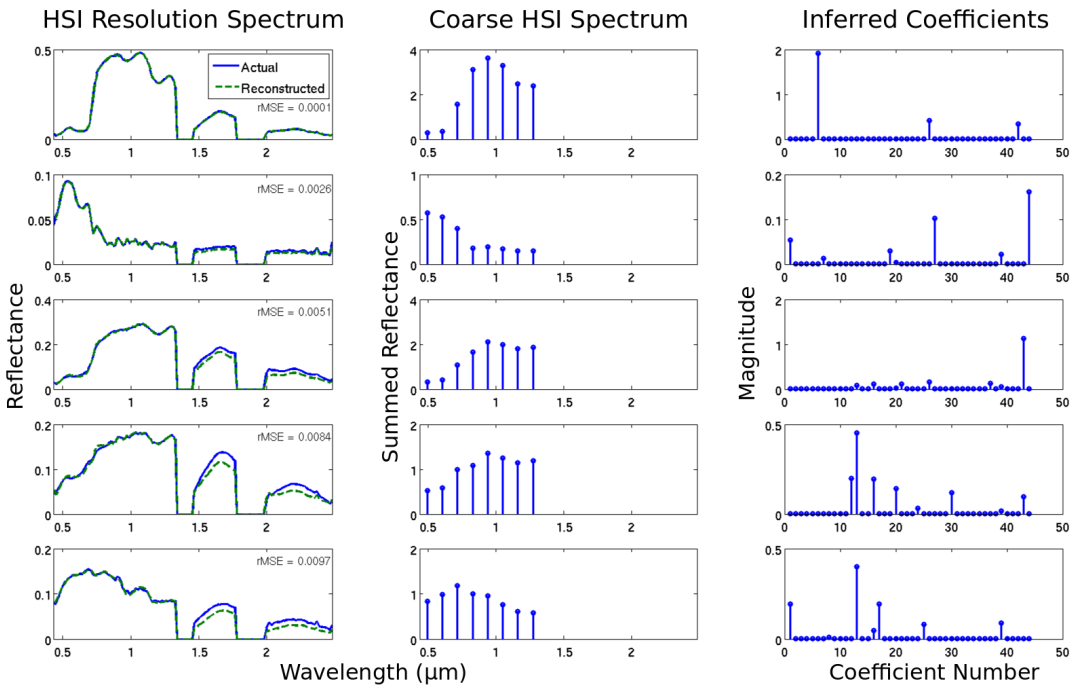

Figure 36: Reconstructing HSI data from simulated MSI measurements using training and testing data collected on the same date. Plots show original HSI spectrum in blue (113 bands), simulated coarse HSI spectrum (8 bands), inferred sparse coefficients, and reconstructed HSI spectrum in green. Examples were selected to illustrate a range of recovery performance, from examples of the best recovery on top to examples of the worst recovery on the bottom.

SD and DD testing datasets in the simulation, with Figures 36 and 37 showing example reconstructions and Table 3 reporting average reconstruction results. While the overall performance does suffer compared to the previous experiment when the whole spectral range was measured, the HSI spectra are again recovered with low error overall: less than 1.28% for the SD dataset, and less than 2.47% error for the DD dataset. As expected from the previous simulation, the lower wavelengths can be reconstructed very well. As might be expected because no data was collected in the higher wavelength range, the recovery in these spectral bands can suffer from higher errors even despite getting the general shape correct. Table 3 also shows that overall, both the learned and exemplar dictionaries have approximately the same mean relative error in this setting. However, the distribution of the

relative errors over the test pixels is more tightly peaked about the origin for the learned dictionary, with a median relative error approximately a third of that for the exemplar dictionary. This indicates that while most test pixels were recovered better with the learned dictionary, there were a minority of pixels that suffered more egregious errors than seen with the exemplar dictionary.

Though the results of the high-resolution reconstructions given above are very encouraging, as with any engineering application it is important to characterize what causes variations in the performance. Figure 38 shows a more detailed analysis of the errors for the worst performing case in the above simulations: using simulated MSI data with a dictionary that was learned on data taken on a different date from the test data. This analysis quantifies the observation that the better the model is at fitting the data, the better we expect the resulting algorithm to perform. Specifically, we group the pixels in the test dataset into three groups based on the (normalized) sparsity of their resulting inferred coefficients (i.e., how well the data point is fit by a sparsity model) measured by $\|\boldsymbol{a}\|_1/\|\boldsymbol{a}\|_2$. The clear trend is that the performance in this task is strongly dependent on how amenable that pixel is to admitting a sparse decomposition. Fortunately, only a small fraction of the data (less than 9%) falls into the worst performing category. Currently we have not found any quantitative correlations between material classes and model fit, but anecdotally we observe that classes such as pine trees and water appear prevalent among the pixels with the lowest rMSE in the reconstruction task, and classes such as mixed vegetation and mud are more prevalent in the outliers that have higher rMSE. Of course, an interesting topic of future study would be to understand more precisely how to modify the model to improve the fit with the current outliers (and subsequently the performance on the current task).

We note that there are many other linear inverse problems that may be of interest, including other methods for reducing data acquisition resources. For example, in the field of compressed sensing [56], a sparsity model is also assumed and data is measured by using

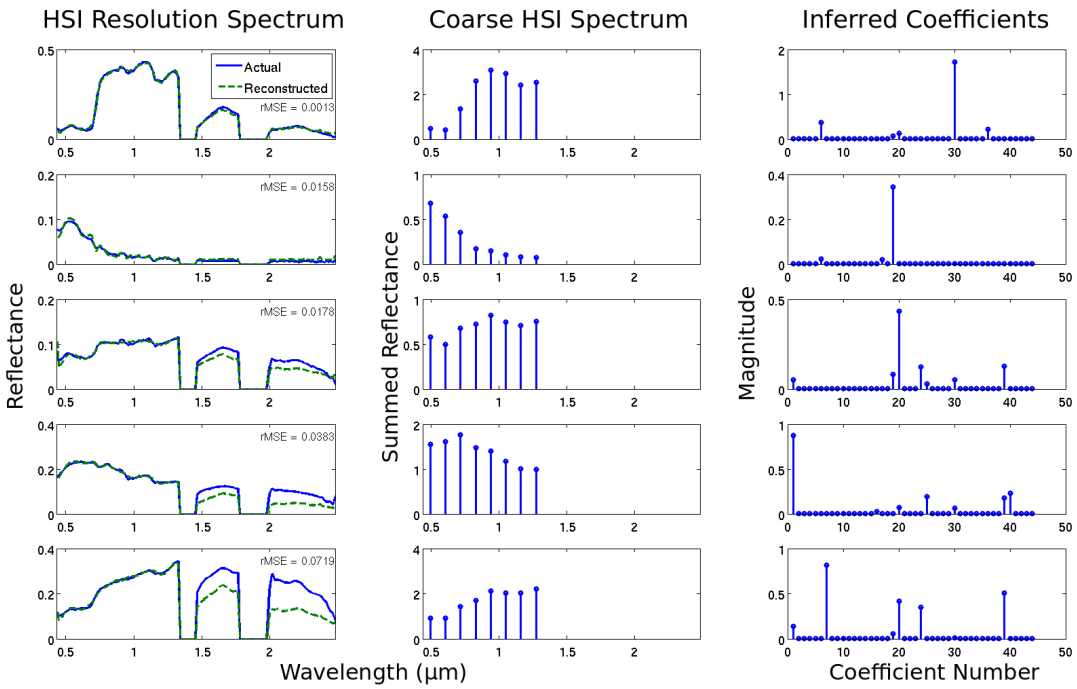

Figure 37: Reconstructing HSI data from simulated MSI measurements using training and testing data collected on different dates (in different seasons). Plots show original HSI spectrum in blue (113 bands), simulated coarse HSI spectrum (8 bands), inferred sparse coefficients, and reconstructed HSI spectrum in green. Examples were selected to illustrate a range of recovery performance, from examples of the best recovery on top to examples of the worst recovery on the bottom.

a coded aperture that forms each measurement by taking a (generally random) linear combination of the input data. In this case, the original data is recovered by solving the same optimization problem as in (67). Indeed, similar acquisition strategies have already been implemented in novel HSI sensors [171, 172]. Looking carefully, the only difference between the compressed sensing strategy and the approach presented above is in the choice of $\boldsymbol{B}$. The "blurring" choice of $\boldsymbol{B}$ in our experiments should actually result in a more difficult reconstruction problem than when $\boldsymbol{B}$ is chosen to be a random matrix because the introduction of randomness will tend to improve the conditioning of the acquisition operator. We have performed similar simulations to the ones above (not shown) using $\boldsymbol{B}$ drawn randomly and independently from a Bernoulli distribution, and the results indicate that recovery with

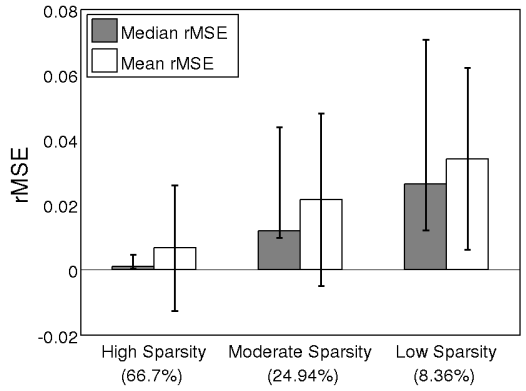

Figure 38: The reconstruction errors when inferring HSI-resolution data from simulated MSI measurements are closely related to the normalized sparsity $\|\boldsymbol{a}\|_1/\|\boldsymbol{a}\|_2$ of the coefficients. The mean and median errors are shown for 3 categories measuring the sparsity model fit: High Sparsity represents an excellent model fit (normalized sparsity is between 1 and 1.92), Moderate Sparsity represents a good model fit (normalized sparsity is between 1.92 and 2.3), and Low Sparsity represents only a fair model fit (normalized sparsity is above 2.3). The data shown is for the reconstructed pixels in the worst performing scenario in our simulations (test pixels from the August 2001 dataset and dictionaries learned from the October 2001 dataset), and the percentage of pixels falling into each category are displayed below the category labels. The error bars of the mean rMSE represent the standard deviation and the error bars on the median rMSE represent the 25^{th} and 75^{th} percentiles. The differences between these two indicates that the reconstruction errors are tightly packed for the data points with low normalized sparsity with a few outliers, and spread out for points with higher normalized sparsity.

similar accuracy is also possible when using this learned dictionary.

5.3.3 Supervised classification

Clearly one of the most important HSI applications is classifying the dominant materials present in a pixel [160, 161, 173]. Because sparse coding is a highly nonlinear operation that appears to capture different spectral features by using different dictionary elements (and not just changing the coefficient values on those elements), we suspect that performing classification on the sparse coefficients can improve HSI classification performance compared to classification on the raw data (or other dimensionality reduced representations such as PCA). Intuition for this approach comes from the well-known idea in machine

learning that expanding a data representation with a highly nonlinear kernel can serve to separate the data classes and make classification easier (especially with a simple linear classifier). Indeed, several researchers have reported that sparse coding in highly overcomplete learned dictionaries (which is a highly nonlinear mapping) does improve classification performance [174, 175].

To gain further intuition, consider a very simple classifier based on finding the maximum sparse coefficient for each pixel in the scene. This sparse decomposition with one coefficient can be thought of as a type of vector quantization (VQ) [176], and the coefficient index can be used as a rough determination of the class of the pixel. Figure 39 shows a segment of the Smith Island dataset, where each pixel is independently unmixed and colored according to the index of the maximum sparse coefficient representing that pixel.[6] Relevant environmental features such as tree lines and sandbanks are clearly distinct, indicating a correlation between the most active dictionary element and the material in the image. Additionally, variations within a class can be captured by different coefficients. For example, different water characteristics are clearly visible, including depth changes due to sandbars (the orange stripes in the left side of the image) and areas with submerged nets (the red stripes offshore by the sandbanks).

While the simple demonstration in Figure 39 is an encouraging illustration, this approach clearly going to underperform compared to a classification scheme that includes information from all of the coefficients simultaneously. To demonstrate the utility of sparse coefficient representations using learned dictionaries for classification, we performed several classification tests on the Smith Island dataset using Support Vector Machines (SVMs) and verifying the results with ground truth labels. SVMs [177] are a widely used supervised learning technique capable of performing multi-class classification. Specifically, we use the C-SVM algorithm (implemented in the freely available `libsvm` package [178]) with a linear kernel.

[6]The colors in Figure 39 are assigned to give as much visual distinction as possible between elements that are physically adjacent, but have no other meaning.

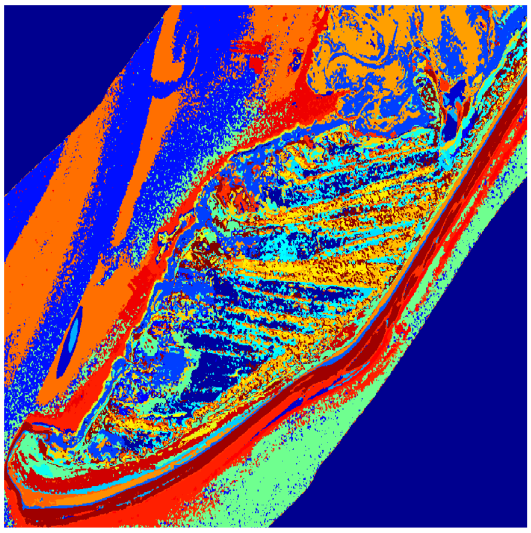

Figure 39: Vector Quantization classification of the scene. The color in each pixel indicates which dictionary element had the largest coefficient value in the sparse code for that pixel. Distinct shapes consistent with known material structures from the ground truth data (e.g., sand bars and tree lines) can be easily seen.

There are two potential factors to consider when performing supervised classification: overall performance (i.e., classification error) and classifier complexity. While classification error on a test dataset is an obvious performance metric of interest, classifier complexity is also an important aspect to consider. For a fixed performance rate, less complex classifiers take less computation time (which is important in large datasets), and are typically less prone to over-fitting during the training (which may lead to better generalization beyond the training data). With linear kernels, the only parameter of the C-SVM algorithm is the cost variable C which controls the complexity of the classifier by changing the cost of the wrongly classified points in the training process. We sweep C over a range of values from 1 to 10000 and observe the probability of error and classification time using the raw HSI data, reduced dimensionality data using PCA, and sparse coefficient representations for the learned, exemplar and random dictionaries discussed earlier. For each value of C, we performed 20 trials where each trial consists of selecting a subset of 17 pixels from the

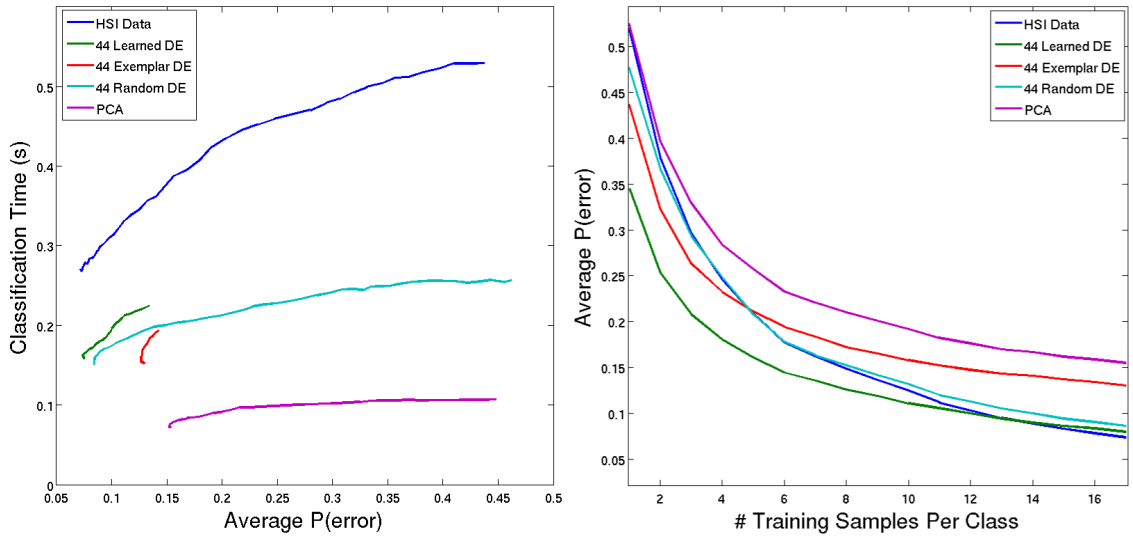

Figure 40: Classification on 22 material classes in the Smith Island dataset. (Left) Average classification error plotted as a function of average classification time (as a proxy for classifier complexity) as the complexity parameter of the SVM is varied. Using coefficients from a sparse code in a learned dictionary as input to the SVM performs essentially as well as using the raw data, but with a classifier 30% less complex. (Right) Average classification error as a function of the training dataset size for each class. The power of the lower complexity classifier is demonstrated in the ability to generalize better, with sparse coefficients in the learned dictionary clearly showing better performance for the very small training sets.

labeled data for each of the 22 classes to train a new SVM classifier and then testing the classification performance on the remaining labeled data withheld from the SVM training.[7] We average over all trials and all 22 classes to find the average classification error and average classification time (as a proxy for classifier complexity). Figure 40 shows the changes in classification time and probability of classification error.

There are three interesting things to note about the results in Figure 40. First, while the raw data achieved the lowest overall error for the range of C tested (P(error)=0.0721),

[7] We choose a training set size of 17 because we want the same amount of training data per class, and the smallest class has 18 labeled samples (leaving one testing pixel for the cross-validation). Average classification performance can be improved significantly on this dataset when larger training samples are used (but at the expense of consistent training set sizes per class).

the sparse coefficients in the learned dictionary are nearly as good (P(error)=0.0736) using a much simpler classifier that operates $\sim$ 30% faster than the SVM on the raw data.[8] Second, while PCA reduces the classification time farther then the other approaches due to its extremely low dimensionality (4 principle components), it performs significantly worse than the raw data or the sparse coefficients. Third, using sparse coefficients in the random dictionary surprisingly performs better (P(error)=0.0838) than sparse coefficients in the exemplar dictionary (P(error)=0.1262), despite having no apparent relevance to material spectra in the scene. While this is counter-intuitive, other recent results have shown that projection onto random dictionaries can be a way to preserve information useful for classification [174, 179], and it is likely that these dictionaries cover the signal space better than random pixels drawn from the labeled classes to form the exemplar dictionaries. Despite this, the coefficients of the learned dictionary do perform better than the random dictionary, demonstrating the value of the learning process. Finally, we should note that while we only display average classification errors, there is a wide variety in the per-class classification errors classes (i.e., some classes are inherently very challenging to distinguish because of their similar spectral features [136]). In our observations (not plotted), the relative difficulty of these classes in the classification task is roughly the same in the different data representations.

As mentioned earlier, one advantage of using classifiers with less complexity is that they may generalize better from the training data, especially when the training dataset is very small. We test the generalization ability of the SVM classification approach described above by repeating the experiment with variable sizes for the training dataset, in the extreme case using only one training pixel per class. We performed and evaluated this simulation

[8] We note that in other simulations (not shown), the best classification performance of the SVN does not improve when using a nonlinear kernel such as a radial basis function (though the complexity obviously increases compared to the linear kernel). This indicates that linear decision boundaries are nearly optimal for this particular dataset, and little advantage is gained from a nonlinear mapping of the decision boundary. While in general we would hope to see lower possible classification error when using sparse coefficients, it appears that nonlinear mappings simply do not add much value to the decision boundaries for this particular dataset.

in largely the same manner as described above, fixing $C = 10,000$ to achieve the lowest classification error and conservatively using 50 trials (i.e., random selections of training data for calculating a new SVM) to mitigate the increased result sensitivity due to the low training set size. Figure 40 plots the results, showing that the sparse coefficients in the learned dictionary do in fact generalize better than the other methods, outperforming the other data representations for very small training set sizes (less than 12 training pixels from the total ground truth data).

5.4 Re-weighted ℓ_1 Methods for Spectral Super-resolution

In previous sections we have demonstrated the applicability of sparsity-based methods in spectral super-resolution for HSI [20]. Specifically, by learning a dictionary of spectral signatures that sparsely decompose the spectral response in each pixel, we learn an approximation to the data manifold that captures rich higher-order statistical structure in HSI data. This model can then be used to perform spectral super-resolution from MSI-level data to HSI-level resolution with very high accuracy [20]. In this section we improve on these previous results by proposing a reweighted ℓ_1 spatial filtering algorithm to incorporate spatial regularity to improve spectral super-resolution. This approach closely follows recent work in dynamic filtering where temporal correlations have been used to improve recovery of time-varying signals in a reweighted ℓ_1 framework [10]. The main contribution is to show that more advanced recovery algorithms can produce significant improvements in the spectral super-resolution results for scenes with significant spatial regularity, with most of the improvement coming from pixels that are not well-modeled by a basic sparsity model.

As a first step to improving super-resolution performance, we generalize the sparsity model to allow the SNR for each coefficient to be an unknown parameter that is estimated as part of the inference process. In BPDN (equation (3)), the tradeoff parameter γ depends on the SNR (the ratio of the variance in the sparse coefficients to the noise variance [20]) and is the same for each coefficient. In contrast, the reweighted ℓ_1 (RWL1) framework [64,180]

allows each coefficient $a_{i,j,k}$ its own parameter $\gamma_{i,j,k}$, where a and γ are inferred concurrently. Specifically, RWL1 is equivalent to using the iterative Expectation-Maximization (EM) algorithm to find a joint estimate of a and γ assuming that γ has an i.i.d. Gamma hyperprior distribution. While more technical details of the model and algorithm can be found in [64], the RWL1 algorithm applied to the super-resolution problem can be stated succinctly as alternating a weighted BPDN optimization and an analytic update to the weights until convergence:

$$\widehat{\boldsymbol{a}}_{i,j}^{n} = \arg\min_{\boldsymbol{a}} \left\| \boldsymbol{x}_{i,j} - \boldsymbol{B}\boldsymbol{\Psi}\boldsymbol{a} \right\|_2^2 + \gamma_0 \sum_k \widehat{\gamma}_{i,j,k}^{n-1} a_k,$$

$$\widehat{\gamma}_{i,j,k}^{n} = \frac{\alpha}{\left| \widehat{a}_{i,j,k}^{n} \right| + \beta},$$

where α, β and γ_0 are parameters related to the hyperprior on γ and n is the iteration number.

One way to intuitively understand the RWL1 algorithm is to understand the effect each $\gamma_{i,j,k}$ has on the weighted ℓ_1 optimization problem. Lowering a given $\gamma_{i,j,k}$ value makes it easier for the corresponding coefficient to be activated in the next BPDN iteration. By iteratively recalculating the weights, coefficients that are activated in the initial optimization become more easily activated in future iterations (via smaller weights) and unused coefficients are more difficult to activate in future iterations (via higher weights). Additional literature has linked RWL1 to approximating solutions to ℓ_p regularized least squares problems for $p < 1$ [181] and asymptotic theoretical guarantees in other inverse problems (e.g., compressed sensing) [127].

5.4.1 Reweighted ℓ_1 Spatial Filtering (RWL1-SF)

While spectral statistics are informative enough to perform super-resolution in many cases, spatial regularity can often be leveraged in some types of scenes to improve performance (especially when the sparsity model is not a good fit for a given pixel). Spatial regularity was also used recently in the context of material classification, indicating its utility in HSI [182]. Therefore, as a second step to improving super-resolution performance,

we further generalize the RWL1 model to incorporate spatial information into the inference process. Specifically, in our proposed reweighted ℓ_1 spatial filtering (RWL1-SF), we update the weights for a given coefficient using a combination of information from the previous iteration on neighboring pixels (similar to the reweighted ℓ_1 dynamic filtering algorithm developed in [10]). In this way, even weak evidence from individual pixels in a local neighborhood can be aggregated to improve the inference in cases that would be particularly difficult when just considering individual pixels independently.

To be precise, consider the matrix of all coefficients for the k^{th} dictionary element, $[\boldsymbol{A}_k]_{i,j} = a_{i,j,k}$. In each iteration of RWL1-SF, the weight for the k^{th} coefficient at the pixel in row i and column j is set by a weighted pooling of the previous estimates for the k^{th} coefficient at the neighboring pixels. While there are many potential ways to implement this spatial aggregation and weight updating, in this paper we use a simple linear weighted average:

$$\gamma_{i,j,k} = \frac{\alpha}{\left|[\boldsymbol{K} * \boldsymbol{A}_k]_{i,j}\right| + \beta}$$

where the term $[\boldsymbol{K} * \boldsymbol{A}_k]_{i,j}$ represents the $\{i, j\}^{th}$ term of the kernel $\boldsymbol{K} \in \mathbb{R}^{L \times P}$ convolved with the spatial field of previous estimates for the k^{th} coefficient. Note that while this spatial regularization can accumulate weak evidence spread over several neighboring pixels to perform inference, the model does not force spatial homogeneity so that single-pixel (or sub-pixel) objects are missed. In other words, rather than low-pass filtering the estimates of interest (the $a_{i,j,k}$ variables), the spatial averaging is applied to a second order variable ($\gamma_{i,j,k}$) that simply biases a sparse inference process. In fact, though an explicit test with single-pixel anomalies is beyond the scope of this letter, previous work using this approach for dynamic filtering [10] showed that this method of stochastic filtering is particularly robust to model mismatch.

The kernel $\boldsymbol{K}$ incorporates the knowledge that dependencies should have a limited spatial extent and will be modulated depending on the distance between the pixels, as depicted in Figure 41. The value in the $\{l, p\}^{th}$ entry of $\boldsymbol{K}$ indicates the amount which the

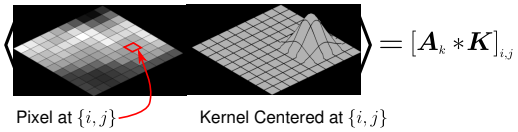

Figure 41: The kernel $\boldsymbol{K}$ determines the influence from neighboring pixels on coefficient inference at a given location. When the $L \times P$ kernel is centered on the $\{i, j\}^{\text{th}}$ pixel it describes the weighted summation of neighboring coefficient estimates that influence the next coefficient estimate in that pixel.

$\{i + l - L/2, j + p - P/2\}^{th}$ element of $\boldsymbol{A}_k$ influences the $\{i, j\}^{th}$ element of $\boldsymbol{A}_k$ in the next iteration of the inference. Typically, the center (0,0) value of $\boldsymbol{K}$ should be unity and the kernel values should taper off towards the edges to represent the decaying dependence with distance. In this work we use the same 5×5 pixel Gaussian kernel shape for all parts of the estimation, but in general each coefficient or pixel location could have a different kernel if there was advanced knowledge of the spatial and spectral dependencies in the data. Indeed, in scenes with very different statistics than the HSI used as an example here (e.g., urban scenes), the spatial regularization process may benefit from a specialized treatment of edges in the image.

5.4.2 Performance Comparisons

We test the performance of RWL1 and RWL1-SF against previous results on segments of HSI from Smith Island, VA. These two HSI images were taken by the PROBE2 sensor on October 18, 2001 and August 22, 2001 and have 113 usable spectral bands spanning the 0.44-2.486μm range (after removal of water absorption bands and applying atmospheric correction to estimate reflectance) and a spatial resolution of approximately 4.5m[9]. We simulate MSI measurements by creating a matrix $\boldsymbol{B}$ to represent a response function that entirely eliminated measurements in higher wavelength regions and pooled the remaining HSI measurements into eight spectral bands shown in Figure 33 (each row of $\boldsymbol{B}$ has ones over bands included and zeros otherwise). We learn a 44-element dictionary $\boldsymbol{\Psi}$ on the

[9]More details about this dataset can be found in [160].

Table 4: Super-resolution from simulated MSI measurements in terms of relative MSE and spectral angle (SA).

	October 18 (Same Day)			
	rMSE		SA (degrees)	
	Mean	Median	Mean	Median
BPDN	2.33%	0.35%	5.838°	3.205°
RWL1	0.85%	0.24%	3.817°	2.683°
RWL1-SF	0.68%	0.23%	3.447°	2.575°
	August 11 (Different Day)			
	rMSE		SA (degrees)	
	Mean	Median	Mean	Median
BPDN	6.25%	6.25%	11.812°	13.587°
RWL1	3.34%	3.02%	8.824°	9.439°
RWL1-SF	2.45%	1.89%	7.492°	7.382°

October 18, 2001 image as in [20], and test recovery on both images. Of particular note is that the two images were taken several months apart, and the statistical changes with the seasonal variations made the recovery of the August image the most challenging test case in prior work [20]. We estimate the original 113 bands from the 8 simulated MSI bands for both images via BPDN, RWL1 and RWL1-SF.

For testing purposes we recover a contiguous 68x288 pixel region (omitting 11 pixels with severe sensor errors) from the Smith Island dataset, shown in Figure 42. This region yielded particularly poor performance when using BPDN for super-resolution in prior work [20]. As shown in Table 4, the previous mean rMSE was 6.3% and the median rMSE was 3.3% for this region on the August image, which is considerably worse than the performance seen on sets of pixels randomly selected throughout the entire image (nearly triple the 2.456% mean and an order of magnitude higher than the 0.1219% median rMSE observed on the full dataset [20]). As stated in [20], BPDN super-resolution resulted in the highest error in portions of the scene that are expected to have more heterogeneous compositions, therefore making the basic sparsity model a poorer fit than it is in more homogeneous regions. To illustrate this, Figure 42 shows the distribution of BPDN reconstruction errors (measured in spectral angle) for the same day dataset, highlighting the difference in performance in distinct regions of heterogeneous materials on the ground. Unsurprisingly,

the higher errors are also concentrated in the HSI spectral bands that are not measured in the MSI data. Previous work [20] shows that if the same number of measurements are taken over the whole HSI spectral range (corresponding to an HSI sensor operating in a lower spectral resolution mode for higher temporal resolution), this ambiguity is reduced and performance increases significantly.

Table 4 provides mean and median recovery results, illustrating significant performance improvements when using RWL1 instead of BPDN, and further substantial improvements when using RWL1-SF. Figure 43 illustrates two example pixels that are representative of the easiest and most challenging performance for the October image. For the best case, the spectra are nearly indistinguishable from the true HSI. For the worst case reconstruction we note that the errors are clearly concentrated in the unmeasured (high wavelength) spectral ranges and that the proposed algorithms make substantial improvements in the recovery over the previous results using BPDN. Figure 44 illustrates that the overall statistics of the data in the August image are also better preserved when using RWL1-SF instead of BPDN, with first four principal components of the reconstructed data (accounting for 99.99% of the energy in the image segment) much more closely approximating the principal components of the HSI when using RWL1-SF.

5.5 Applications to Oceanic Imagery

While previous sections relied on creating simulated MSI measurements from HSI images to test our sparsity-based super resolution techniques, we present here results using geographically co-located images of oceanic water-color. Specifically, we take two images (one taken with the 89-channel HICO sensor and one taken with the 5-channel VIIRS sensor), and resolve the VIIRS spectra to HICO-resolution spectra.

To perform our super-resolution we first learn a dictionary of material spectra via the techniques outlined in [20]. Next we estimate the blurring operator by comparing the

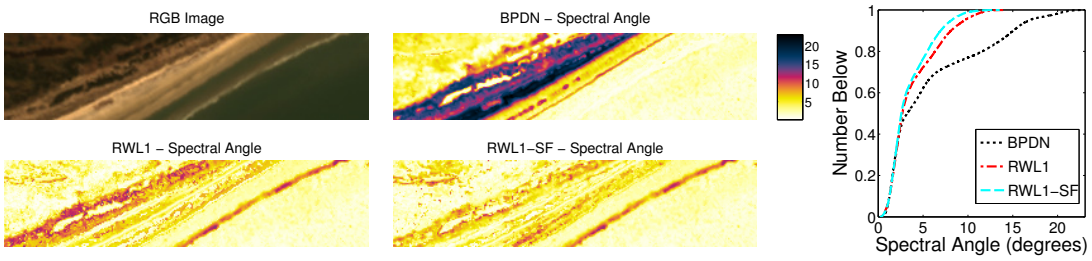

Figure 42: Left: The RGB image of the October region being tested and the heat maps depicting the spectral angle errors throughout the region using BPDN, RWL1 and RWL1-SF. The largest improvements over BPDN occur along the shoreline where the material mixture is very heterogeneous (e.g., water, sand, vegetation) and the sparsity model alone is insufficient. Right: The cumulative distribution function (CDF) of the spectral angle errors. Note that the BPDN CDF has a heavy tail, indicating many pixels with poor performance. RWL1 improves performance significantly. RWL1-SF uses a model of spatial dependence to further reduce the outliers and improve performance, with 90% of the pixels having spectral angle errors less than 6.9536 degrees.

relative signal-to-noise ratios for both the VIIRS and HICO sensors over their respective spectral ranges, which allows us to super-resolve the VIIRS data using our sparsity-based methodology. We validate our super-resolved VIIRS spectra by comparing to high-resolution measurements from the HICO sensor. Figure 46 shows some example recovered spectra. In particular, the most accurate, least accurate and median (typical) recovery, as based on the spectral angle between the recovered spectra and the HICO spectra at the same geographical coordinates, are all shown. With a median spectral angle of 7.43 degrees, we note that the recovered spectra accurately represent the HICO spectra. Figure 47 shows a distribution of errors, and emphasizes that the majority of errors are small with a few outliers. Additionally, we study where and how the super-resolution does not match the HICO data. For example, as depicted in Figure 48, the best matches occurred over water pixels, while the worst matches occurred along the shore. One potential source of this shoreline discrepancy is that along the shore there are typically more materials present, indicating a model mismatch with the sparsity assumption. Another potential source of the mismatch is that although the VIIRS and HICO images were acquired at the same geographical location and at the same date, the images were taken approximately 8 hours apart, indicating that

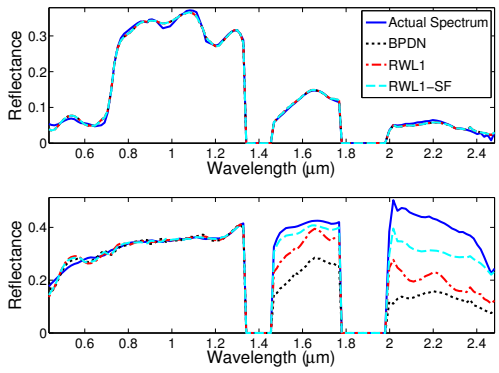

Figure 43: Two example spectra super-resolved from MSI-level data. Top plot is representative of best-case performance and bottom plot is representative of worst-case performance for previous approaches [20]. Note that errors are highly concentrated in unmeasured bands.

tidal changes may have actually changed the shoreline composition from water spectra to land spectra.

Overall, our results indicate that sparsity-based spectral super-resolution techniques can greatly extend the utility of legacy MSI and HSI instruments via post-processing. Additionally, accurate super-resolution could impact future sensor designs by creating options for lighter sensors with reduced transmission bandwidth at the cost of additional computation at base stations.

5.6 Discussion

In this chapter we have shown that a sparse coding model and the dictionary learning approach described in [57] (with minor modifications) can yield valuable representations of HSI data using no *a priori* information about the dataset. The learned dictionary elements resemble many of the spectra corresponding to known material properties in the scene, and the sparse decomposition of the HSI data using this dictionary shows that the variations in the surface properties are often sensibly represented. In particular, in contrast with a typical endmember approach that seeks to contain the HSI data in a convex hull, this learned

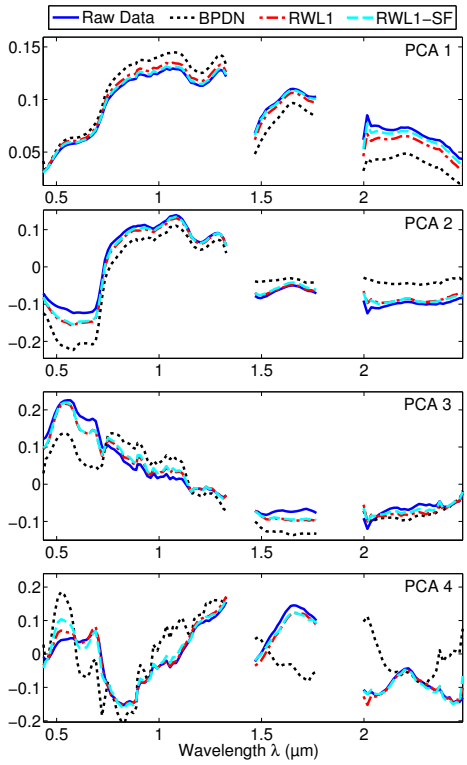

Figure 44: The first four principal components the recovered HSI spectra compared to the principal components of the original HSI data.

dictionary captures nonlinear material variations directly by forming a locally linear approximation to the manifolds observed within a material class.

The learned dictionaries capture many high-order statistics of the data they are learned from, and this representation showes advantages in applications relevant for remote sensing scenarios. For example, when coupled with a linear inverse problem, this learned dictionary demonstrated that HSI-resolution spectra could be recovered with remarkable fidelity from (simulated) spectra collected with just MSI-level resolution. This performance is only possible because the learned dictionaries are capable of effectively capturing the high level of statistical dependencies inherent in HSI data. Furthermore, encouraging results show that the performance on this task is still very good when there is some mismatch in the statistics because the training and testing data was collected at different times (i.e., a different season of the year, with different characteristics in the vegetation and the atmosphere). While

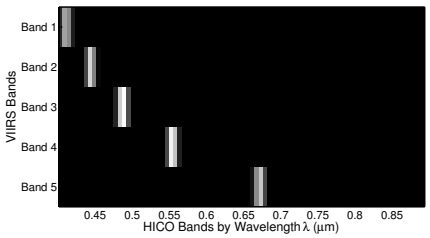

Figure 45: Blurring matrix which transforms HICO spectra into VIIRS spectra. Matrix elements were determined by comparing each sensor's sensitivity function over different wavelength bands.

this reconstruction problem was intended to mimic a realistic and useful data acquisition scenario, we note that this linear inverse problem framework captures many problems of interest (including other acquisition models such as those in compressed sensing [56]).

While initial tests focused on simulated MSI data only, we have also been able to test our techniques on a second dataset that includes co-located HSI and MSI data, allowing us to verify the utility of learned dictionaries for spectral super-resolution. Our analysis concludes that the majority of MSI pixels can accurately be interpolated to HSI-level resolutions, a number of outlier errors can still occur. The geographical distribution of these error, however, lead us to believe that these outlier errors are likely the result of tidal effects, or changes in the scene in the hours between when the HSI and MSI images were taken, although further tests on other co-located images will be required to verify this conjecture.

We note that while the approach using BPDN achieved very good performance for spectral super-resolution in many cases, using enhanced models can substantial improve recovery in the most challenging test cases. The RWL1-SF algorithm leverages both a more advanced sparsity models in each pixel, as well as spatial regularity between pixels. This increased model structure improves super-resolution results significantly, especially in the pixels that were outliers in previous results due to their poor super-resolution performance [20]. Specifically, using additional intra-pixel structure in RWL1 yielded a 35.62% and 16.29% improvement in the mean and median SA, respectively. Incorporating spatial

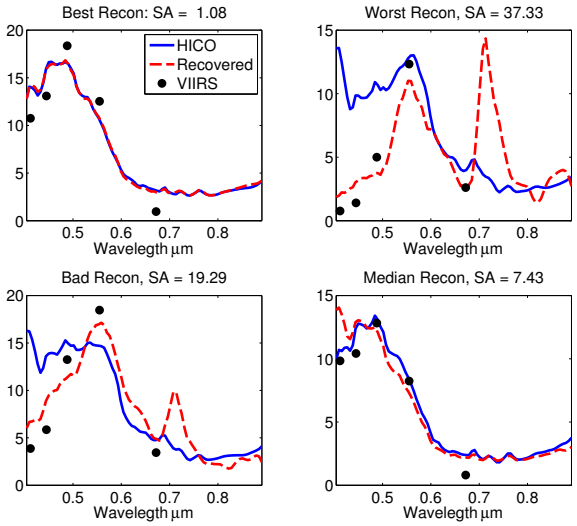

Figure 46: Examples of spectral super-resolution of a VIIRS image taken around the Acqua Alta Oceanographic Tower (AAOT) near Venice, Italy on February 11, 2012. In each figure the black dots represent the five VIIRS measurements, the solid blue lines represent the HICO spectrum captured near that location, and the dashed red line represents the super-resolved VIIRS spectrum. Shown are examples of the best reconstruction (top left) the worst reconstruction (top right), bad reconstruction (bottom left) and median reconstruction (bottom right). As the median reconstruction was fairly accurate, we note that the majority of the super-resolved spectra (in particular water-color pixels) are recovered well.

dependencies in RWL1-SF boosted these results further, giving a total of 40.96% improvement in the mean SA and 19.66% improvement in median SA. In fact, 90% of the recovered pixels in the current dataset had a spectral angle error less than 7 degrees, which is well within the class spectral width of some classifiers currently in use (e.g., 7 degrees to 30 degree in [183]). Again, we note that the presented data includes some of the most challenging problem aspects (i.e., difficult pixels and MSI measurements with no data from some HSI bands).

Finally, we showed that the sparse coefficients from this learned dictionary can form a useful representation for performing classification compared to the raw data, yielding classifiers with less complexity that generalize better when the training dataset size is very small. From these results we can conclude that the sparse coding model is a potentially valuable approach to analyzing HSI data, and the learned dictionaries for this model form a

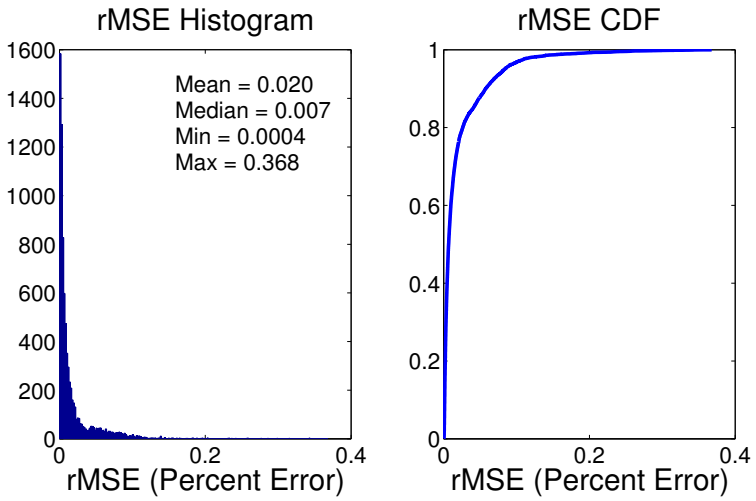

Figure 47: The HICO level super-resolution matches the true HICO spectra. Left: the histogram of rMSE errors, comparing the super-resolved spectra at each geographical location to the HICO spectra at the same location. Since the VIIRs spectra are at a lower spatial resolution than the HICO spectra, we average 3x3 blocks of HICO spectra around the given location and compare the average spectrum to the recovered spectrum.

meaningful representation of the high-order statistics in the HSI data. While this approach shares the same linear model as the common endmember approach for spectral unmixing, the different philosophy of representing the data variations directly appears to have value both in the general understanding of the data and in specific applications. We believe that this exploration (along with the other related results in [149, 151, 156, 170]) demonstrates that more extensive exploration of the utility of this model in HSI is warranted, and improvements in many specific applications are likely. In the future, in addition to more thorough application of these ideas to other datasets, it will be valuable to explore the utility of including increasingly complex models in in the learning process. For example, there may be potential benefits to learning much larger dictionaries than those shown in this work, learning joint spectral-spatial dictionaries, learning dictionaries customized for specific applications (such as in [170]), and learning dictionaries that attempt to explicitly capture features such as correlations between pixels and nonlinear variations within material classes.

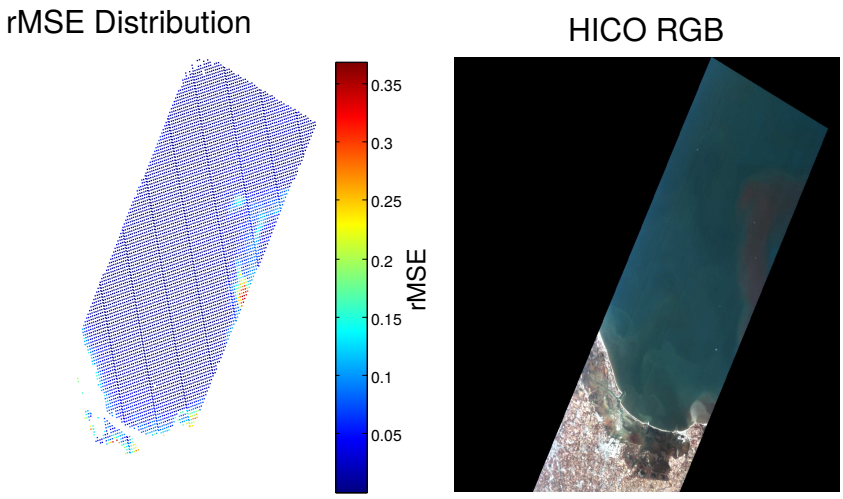

Figure 48: Geographically, the largest errors clustered along shoreline. Right: the RGB color image of the Venice location. Left: the heat map of rMSE errors distributed geographically. Given that the HICO and VIIRS images were taken 8 hours apart, the high errors by the shore-line could indicate errors due to tidal effects.

CHAPTER VI

NETWORK ARCHITECTURES FOR ANALOG OPTIMIZATION[1]

In the final piece of the signal acquisition puzzle we seek efficient estimation implementations. Work along this route has mainly progressed in two directions: digital algorithms which utilize either convex optimization routines or belief propagation, and analog algorithms which can be built on dedicated hardware.

In the digital domain, many efficient implementations of convex optimization algorithms have been specifically tailored to sparse signal estimation. Several interior point methods have been proposed in this area, including ℓ_1-magic [184] and l1-ls [34]. Alternatively, the GPSR algorithm [185] employs a gradient projection approach to solving the BPDN problem. Homotopy (or continuation) methods [186–188] take an entirely different approach, solving a series of optimization problems for a decreasing sequence of tradeoff parameters γ in Equation (3) and utilizing efficient updates to find these sequential solutions. To speed up the recovery process for very large signals, additional work has sought to leverage parallel hardware configurations such as multicore [189] and GPU architectures [190]. While achieving improvements in solution times, neither of these architectures provide favorable scaling properties and it is unclear if they could provide real-time solutions for significantly sized problems. Additionally, neither architecture is appropriate for low-power, embedded computing applications.

Among digital algorithms, the family of iterative thresholding (ITH) methods [124, 191–194] takes a slightly different approach and is closest to dynamical analog systems. These methods iteratively perform gradient-type steps to minimize the cost function (3)

[1]This chapter is in collaboration with Dr. Pierre Garrigues (Section 6.2). ASC, PG, and CJR have contributed equally to this work. Specifically, CJR and PG described the initial problem formulation and derived some of the thresholding functions and ASC derived other thresholding functions and ran significant simulations. More details on this work can be found in [26, 27, 29]

and apply a thresholding function to enforce the sparsity constraints. One member of the ITH family that specifically solves the BPDN problem of Equation 3 is the iterative soft-thresholding algorithm (ISTA). ISTA is essentially a proximal gradient algorithm [195], however for BPDN-type problems, the proximal projection reduces to a computationally cheap soft thresholding operation. Specifically, ISTA alternates between updating a residual of the sparse coefficient vector $\boldsymbol{a}$ using the ℓ_2 norm and soft thresholding the coefficients to enforce sparsity. This optimization program can be written as the following iterative procedure over an algorithmic time l,

$$\begin{aligned} \boldsymbol{u}^{l+1} &= \widehat{\boldsymbol{a}}_n^l + \eta \boldsymbol{\Psi}^T \boldsymbol{\Phi}^T \left(\boldsymbol{y} - \boldsymbol{\Phi} \boldsymbol{\Psi} \widehat{\boldsymbol{a}}^l \right) \\ \widehat{\boldsymbol{a}}^{l+1} &= T_\gamma(\boldsymbol{u}^{l+1}), \end{aligned} \tag{69}$$

where $\boldsymbol{u}$ is the un-thresholded version of the signal that gets updated by the error residual at each algorithmic time-step l, and η is the algorithm's step size. Additionally, approaches based on linearized Bregman iterations have been shown to have update steps that have a similar form [154].

In this proposal we are principally concerned with analog dynamical systems that can solve sparsity-inducing optimization problems. Analog signal processing is of particular interest here since analog systems can run orders of magnitude faster and in low-power conditions [27]. The most prevalent analog algorithm which has been devised for sparse signal estimation is the locally competitive algorithm (LCA) [35]. This system was designed as a Hopfield-type network which updates a series of node values $\boldsymbol{u} \in \mathbb{R}^{N_2}$ according to a continuous-time dynamical equation:

$$\dot{\boldsymbol{u}}_t = -\frac{1}{\tau} \left(\boldsymbol{u}_t + \boldsymbol{\Psi}^T \boldsymbol{\Phi}^T \boldsymbol{y} - (\boldsymbol{\Psi}^T \boldsymbol{\Phi}^T \boldsymbol{\Phi} \boldsymbol{\Psi}_t - \boldsymbol{I}) \boldsymbol{a}_t \right), \tag{70}$$

where τ is the system's time constant and the sparse representation $\boldsymbol{a}$ is related to the node values by a non-linear thresholding function:

$$\boldsymbol{a}(t) = T_\gamma(\boldsymbol{u}(t)), \tag{71}$$

where the parameter γ is dependent on the cost function. The estimate of the sparse representation is then $\widehat{\boldsymbol{a}} = \lim_{t\to\infty} \boldsymbol{a}(t)$. In the case of the BPDN cost of Equation (3), T_γ is the well-known soft thresholding function. Related work has shown various properties of this system (i.e., global convergence given certain conditions on $T_\gamma(\cdot)$ [196] and hardware implementations [27]). The majority of this work, however, focuses on specific cost functions – in particular the ℓ_1 cost. Showing that other sparsity-inducing cost functions could be likewise implemented in an analog architecture could allow those estimators to also be used in low-power, real-time applications.

6.0.1 Sparse Coding

In the sparse coding problem, we use probabilistic inference to find the smallest number of causes for an observed signal under a linear generative model

$$\boldsymbol{x} = \Phi\boldsymbol{a} + \boldsymbol{\epsilon}, \tag{72}$$

where $\boldsymbol{x} \in \mathbb{R}^M$ is the observed signal, $\boldsymbol{a} \in \mathbb{R}^N$ is the coefficient vector, $\Phi \in \mathbb{R}^{M\times N}$ is the dictionary of causes, and $\boldsymbol{\epsilon}$ is Gaussian noise. The coefficient vector is said to be sparse as we seek a solution with relatively few non-zero entries. The coefficients $\boldsymbol{a}$ are generally inferred via MAP estimation, which results in solving a non-linear optimization problem

$$\min_{\boldsymbol{a}} \; E = \frac{1}{2}\left\|\boldsymbol{x} - \Phi\boldsymbol{a}\right\|_2^2 + \lambda\widetilde{C}\left(\boldsymbol{a}\right), \tag{73}$$

where $\widetilde{C}\left(\cdot\right)$ is a cost function penalizing $\boldsymbol{a}$ based on its fit with the signal model, and λ is a parameter denoting the relative tradeoff between the data fidelity term (i.e., MSE, which arises from the log likelihood of the Gaussian noise) and the cost function. The cost function is the non-linear sparsity-inducing regularization term, corresponding to the log prior of the data model. More details about the formulation of this problem in the Bayesian inference framework can be found in [57]. Basic signal models frequently assume independence among the elements of $\boldsymbol{a}$, resulting in a cost function that separates into a sum of individual costs $\left(\text{i.e., } \widetilde{C}\left(\boldsymbol{a}\right) = \sum_k C\left(a_k\right)\right)$. One common example is the ℓ_p norm, defined as $\widetilde{C}\left(\boldsymbol{a}\right) = \left\|\boldsymbol{a}\right\|_p^p = \left(\sum_i a_i^p\right)$.

6.0.2 Dynamical systems for ℓ_1 minimization

As mentioned above, recent work in computational neuroscience has shown that the LCA dynamical system provably solves the optimization programs in (73) and are efficient for solving the non-smooth problems of interest in sparse approximation. The LCA [35] architecture is comprised of a network of analog nodes being driven by the signal to be approximated. Each node competes with neighboring nodes for a chance to represent the signal, and the steady-state response represents the solution to the optimization problem.

The LCA is a specific type of Hopfield neural network, which have a long history of being used to solve optimization problems [197]. The LCA is a neurally plausible architecture, consisting of a network of parallel nodes that use computational primitives that are well-matched to individual neuron models. In particular, each node consists of a leaky integrator and a non-linear thresholding function, and it is driven by both feedforward and lateral (inhibitory and excitatory) recurrent connections. This architecture has been implemented in neuromorphic hardware, both as a purely analog system [27] and by using integrate and fire spiking neurons for each node [198]. We also note that other types of network structures have also been proposed recently to approximately solve specific versions of the sparse approximation problem [199–202].

Specifically, the k^{th} node of the LCA is associated with $\boldsymbol{\phi}_k$, the k^{th} column of Φ. Without loss of generality, we assume each column has unit norm. This node is described at a given time t by an internal state variable $u_k(t)$. The coefficients $\boldsymbol{a}$ are related to the internal states $\boldsymbol{u}$ via an activation (thresholding) function $\boldsymbol{a}(t) = \widetilde{T}_\lambda(\boldsymbol{u}(t))$ that is parametrized by λ. In the important special case when the cost function is separable, the output of each node k can be calculated independently of all other nodes by a pointwise activation function $a_k(t) = T_\lambda(u_k(t))$. Individual nodes are leaky integrators driven by an input proportional to $\langle \boldsymbol{\phi}_k, \boldsymbol{x} \rangle$, and competition between nodes occurs via lateral connections that allow highly

active nodes to suppress nodes with less activity. The dynamics for node k are given by:

$$\dot{u}_k(t) = \frac{1}{\tau}\left[\langle \boldsymbol{x}, \boldsymbol{\phi}_k \rangle - u_k(t) - \sum_{\substack{j=1 \\ j \neq k}}^{N} \langle \boldsymbol{\phi}_k, \boldsymbol{\phi}_j \rangle a_j(t)\right], \tag{74}$$

where τ is the system time constant. In vector form, the dynamics for the whole network are given by:

$$\dot{\boldsymbol{u}}(t) = \frac{1}{\tau}\left[\Phi^t \boldsymbol{x} - \boldsymbol{u}(t) - \left(\Phi^t \Phi - I\right) \boldsymbol{a}(t)\right]. \tag{75}$$

In [35] it was shown that for the energy surface E given in (73) with a separable, continuous and piecewise differentiable cost function, the path induced by the LCA (using the outputs $a_k(t)$ as the optimization variable) ensures $\frac{dE(t)}{dt} \leq 0$ when the cost function satisfies:

$$\lambda \frac{dC\left(a_k\right)}{da_k} = u_k - a_k = u_k - T_\lambda\left(u_k\right) = T_\lambda^{-1}(a_k) - a_k, \tag{76}$$

where $T_\lambda\left(u_k\right)$ is non-decreasing. We use the notation $T_\lambda^{-1}(u_k)$ for convenience when the activation function is invertible, but this invertibility is not strictly required (i.e., the relationship in (76) involving just $T_\lambda\left(u_k\right)$ is sufficient). The same arguments also extend to the more general case of non-separable cost functions, ensuring $\frac{dE(t)}{dt} \leq 0$ when

$$\lambda \nabla_{\boldsymbol{a}} \widetilde{C}\left(\boldsymbol{a}\right) = \boldsymbol{u} - \boldsymbol{a} = \boldsymbol{u} - \widetilde{T}_\lambda\left(\boldsymbol{u}\right) = \widetilde{T}_\lambda^{-1}\left(\boldsymbol{a}\right) - \boldsymbol{a}. \tag{77}$$

Recent followup work [203] establishes stronger guarantees on the LCA, specifically showing that this system is globally convergent to the minimum of E (which may be a local minima if $C\left(\cdot\right)$ is not convex) and proving that the system converges exponentially fast with an analytically bounded convergence rate.

The relationship in (76) requires cost functions that are differentiable and activation functions that are invertible. However, the cost function for BPDN (the ℓ_1 norm) is non-smooth at the origin and the most effective sparsity-promoting activation functions will likely have non-invertible thresholding properties. In these cases, one can start with a smooth cost function that is a relaxed version of the desired cost and calculate the corresponding activation function. Taking the limit of the relaxation parameter in the activation

function yields a formula for $T_\lambda(\cdot)$ that can be used to solve the desired problem. Specifically, in the appendix we use the log-barrier relaxation [204] to show that the LCA solves BPDN when the activation function is the well-known soft thresholding function:

$$C(a_k) = |a_k| \quad \Longleftrightarrow \quad a_k = T_\lambda(u_k) = \begin{cases} 0 & |u_k| \leq \lambda \\ u_k - \lambda \text{sign}(u_k) & |u_k| > \lambda \end{cases}.$$

Similarly, the LCA can find a local minima to the non-convex optimization program that minimizes the ℓ_0 "norm" of the coefficients (i.e., number of non-zeros) by using the hard thresholding activation function [35]:

$$C(a_k) = I(a_k \neq 0) \quad \Longleftrightarrow \quad a_k = T_\lambda(u_k) = \begin{cases} 0 & |u_k| \leq \lambda \\ u_k & |u_k| > \lambda \end{cases},$$

where $I(\cdot)$ is the standard indicator function.

6.1 CS Recovery via the LCA

In this section, we demonstrate the possible performance of the LCA on large-scale CS recovery problems by simulating the ideal dynamical system (described in equations (70) and (71)), illustrating that the potential benefits justify continued efforts to scale up the current implementation. Specifically, we show the soft-thresholding cost function solves the BPDN problem of Equation (3) and then provide simulations that analyze the LCA's solution quality. In the first set of simulations (Sections 6.1.2 and 6.1.3), we use synthetic stylized data to thoroughly explore the solution quality and solution times with (simulated) analog and digital approaches for $N = 1000$. In the second set of simulations (Section 6.1.4), we use very high dimensional MRI data to show performance on a large-scale problem of practical importance.

6.1.1 BPDN optimization through the LCA

To show that the LCA with soft-thresholding solves the BPDN equation, we first rewrite the desired BPDN problem in an extended formulation to make the variables non-negative.

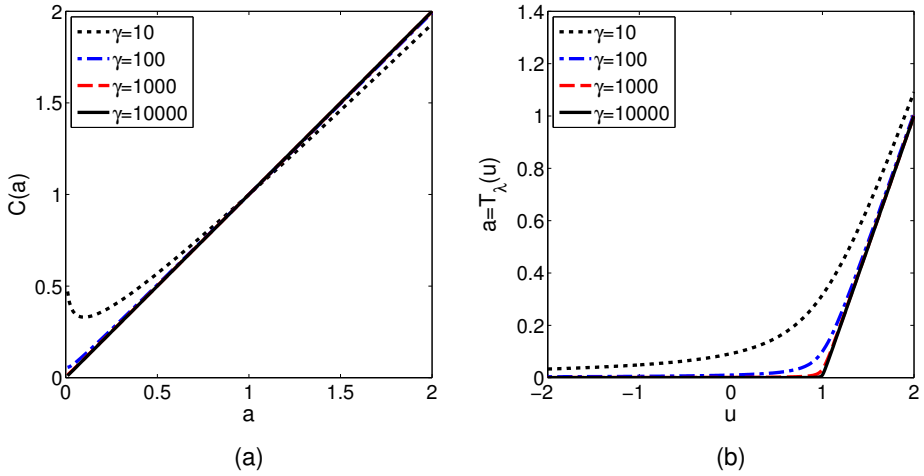

Figure 49: Log barrier relaxations of BPDN. (a) The cost function approaches the ideal ℓ^1 norm as the relaxation parameter is increased. (b) In a similar way, the nonlinear activation function derived for the LCA approaches the ideal soft-thresholding operator as the relaxation parameter is increased.

Define a new $M \times 2N$ matrix through the concatenation operation $\widetilde{\Phi} = [\Phi \ -\Phi]$. Similarly define a vector $\boldsymbol{z} = [z_+ \ z_-]$ of length $2N$ such that $z_i \geq 0$ and $\boldsymbol{a} = z_+ - z_-$. Essentially $\boldsymbol{z}$ represents the original variables $\boldsymbol{a}$ by separating them into two subvectors depending on their sign. We can then write a constrained optimization program that is equivalent to BPDN:

$$\min_z \frac{1}{2} \left\| \boldsymbol{x} - \widetilde{\Phi} \boldsymbol{z} \right\|_2^2 + \lambda \sum_{k=1}^{2N} z_k \qquad \text{s.t.} \quad z_k \geq 0. \tag{78}$$

This reformulation is a standard way to show that ℓ^1 cost penalties are equivalent to a linear function in a constrained optimization program. One can then apply the standard log-barrier relaxation to convert the program in (78) to an approximately equivalent unconstrained program:

$$\min_z \frac{1}{2} \left\| \boldsymbol{x} - \widetilde{\Phi} \boldsymbol{z} \right\|_2^2 + \lambda \sum_{k=1}^{2N} z_k + \left(\frac{1}{\gamma} \right) \sum_{k=1}^{2N} \log(z_k). \tag{79}$$

As $\gamma \to \infty$, this program approaches the desired program (78). This relaxation strategy underlies an interior point algorithm (called the barrier method) for solving convex optimization programs, where (79) is repeatedly solved with increasing values of γ [204].

Note that the relaxed problem in (79) fits the form of the general optimization program

stated in (73) with the differentiable cost function $C(z_k) = z_k - \frac{\log(z_k)}{\gamma\lambda}$. For a fixed value of γ, this cost function can be differentiated and used in the relationship given in (76) to solve for z_k in terms of u_k to find the corresponding invertible activation function:

$$z_k = T_\lambda(u_k) = \frac{1}{2}\left(\sqrt{\frac{4+\gamma(\lambda-u_k)^2}{\gamma}} - (\lambda - u_k)\right).$$

Finally it is straightforward to show that in the relaxation limit ($\gamma \to \infty$) where the program in (79) approaches BPDN, the desired activation function becomes the soft-thresholding function:

$$\begin{aligned}\lim_{\gamma\to\infty} \frac{1}{2}\left(\sqrt{\frac{4+\gamma(\lambda-u_k)^2}{\gamma}} - (\lambda-u_k)\right) &= \frac{1}{2}\left(\sqrt{(\lambda-u_k)^2} - (\lambda-u_k)\right) \\ &= \begin{cases} 0 & \text{when } u_k \le \lambda \\ u_k - \lambda & \text{when } u_k > \lambda \end{cases}.\end{aligned}$$

To illustrate the convergence of this relaxation to the desired ℓ^1 cost function and the corresponding soft-threshold activation function, Figure 49 plots $C(\cdot)$ and $T_\lambda(\cdot)$ in this relaxed problem for several values of γ. Note that in the extended formulation of BPDN given in (78), the variables occur in pairs where where only one of them can be nonzero at a time. Because the activation function is zero for all state values with magnitude less than threshold, it is possible to represent each of these pairs of variables in one LCA node that can take on positive and negative values and where the activation function is a two-sided soft-thresholding function (thereby reducing the number of nodes back down to N).

6.1.2 LCA solution quality

To begin, we investigate the quality of simulated LCA solutions on CS recovery problems with synthetic data to verify that they are comparable to standard digital algorithms. While the LCA system is proven to converge asymptotically to the unique BPDN solution, the approximate solution achieved by any algorithm in finite time can have different characteristics depending on the particular solution path. In the general problem setup, the unknown

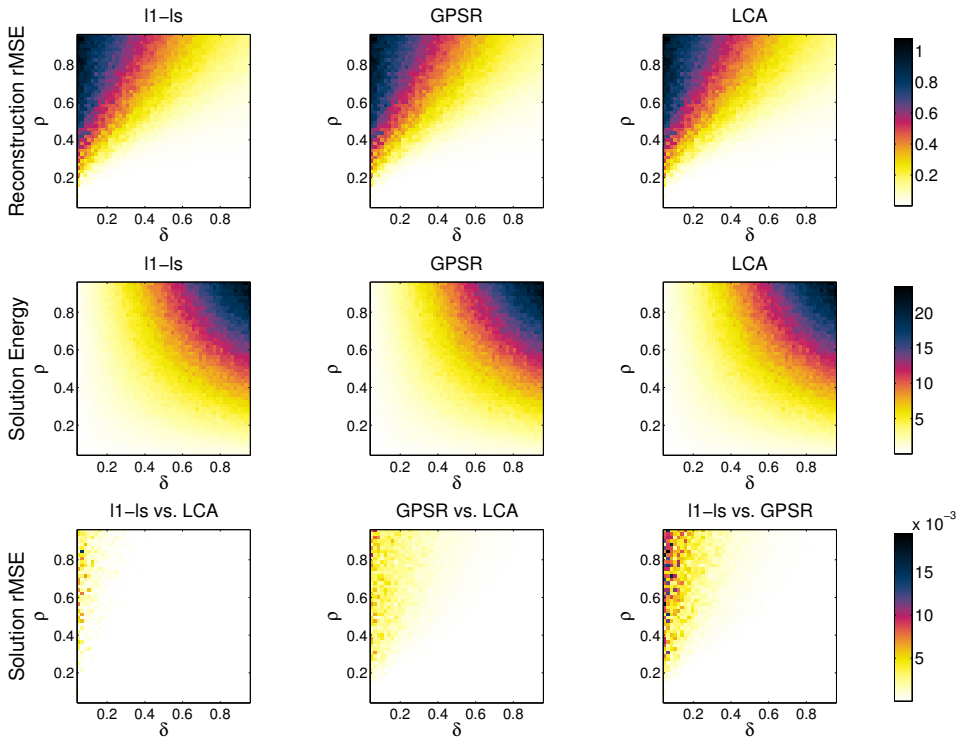

Figure 50: The solution quality of the simulated LCA on a compressed sensing recovery task is comparable to the standard digital solvers GPSR and l1-ls. The top row plots the relative MSE of the estimated signal for synthetic data, with indeterminacy of the system indexed by $\delta = M/N$, and the sparsity of the system with respect to the number of measurements indexed by $\rho = S/M$. The middle row plots the value of the BPDN objective function at the solutions. The bottom row plots the relative MSE in the solutions between the solvers, indicating the the differences in the LCA solutions are within the normal range of differences between the digital algorithms themselves. Note that all solvers demonstrate more variability in regions where the problems are more difficult and signal recovery cannot be performed well.

signal $a_0 \in \mathbb{R}^N$ is S-sparse and is observed through $M < N$ Gaussian random projections, $y = \Phi a_0 + \nu$, where ν is additive Gaussian noise. We compare the simulated performance of the LCA at recovering a_0 BPDN against the interior-point method l1-ls [34] and the gradient projection method GPSR [185]. This investigation will address two main questions. First, are the solutions produced by the simulated LCA as accurate as the digital comparison cases? Second, what solution times are possible in the simulated LCA?

We draw the nonzero coefficients of a_0 using a Gaussian distribution with variance 1

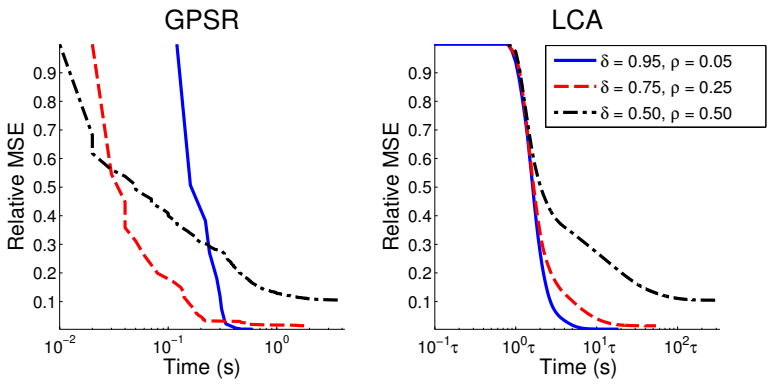

Figure 51: Temporal convergence of the simulated LCA compared to GPSR. The plot shows the relative MSE of the signal recovery as a function of time for sample trials (N=1000) from the results in Fig. 52 using GPSR (left) the simulated LCA (right). The convergence behavior is approximately the same, with harder problems taking both algorithms longer and decreasing the fidelity of the recovery. For the easy and medium difficulty problems where BPDN recovers the signal with good fidelity, GPSR takes 0.1-1 seconds to converge and the simulated LCA takes $10^1\tau$-$10^3\tau$ seconds to converge. For reasonable values of τ, the LCA solution times can still be as low as 10μs, supporting datarates of up to 100 kHz

and we draw the locations from a uniform distribution. The choice of regularization parameter λ depends on the variance of the additive noise ν which is not necessarily known a priori. We have empirically observed that $\lambda = .01\|\Phi^T y\|_\infty$ gives good performance in this task when the noise variance is 10^{-4}. Additionally, we observe that as with many other algorithms, implementing a continuation method by gradually decreasing λ (similar to that used in FPC [193]) also improves convergence time in the LCA. Specifically, we initialize $\lambda = \|\Phi^T y\|_\infty$ and allow a multiplicative decay of 0.9 at each iteration of the simulation until λ reaches the desired value given above. Although the implementation of the current hardware only supports a constant threshold value over time, inclusion of a decaying threshold is possible by having temporally changing threshold currents I_{th} at the threshold units. To ensure that the comparison among the algorithms is fair, we use the same stopping criterion for convergence based on the duality gap upper bound proposed in [34].

To explore solution quality we display the results of solving the CS recovery optimizations using plots inspired by the phase plots described by Donoho & Tanner [54]. We

parameterize the plots using the indeterminacy of the system indexed by $\delta = M/N$, and the sparsity of the system with respect to the number of measurements indexed by $\rho = S/M$. We vary δ and ρ in the range $[.1, .9]$ using a 50 by 50 grid. For a given value (δ, ρ) on the grid, we sample 10 different signals using the corresponding (M, N, S) and recover the signal using BPDN. We compare the results of the simulations by displaying in the top row of Fig. 50 a phase plot for each algorithm, where the color code depicts the average relative MSE of the CS recovery for each algorithm (calculated by $\left\|\widehat{a} - a_0\right\|_2^2 / \|a_0\|_2^2$). In a similar vein, the middle row of Fig. 50 shows the energy function (i.e., the BPDN objective function) evaluated at the solution, $0.5 \left\|y - \mathbf{\Phi}\widehat{a}\right\|_2^2 + \lambda \left\|\widehat{a}\right\|_1$.

The near identical plots for the two metrics above demonstrate that the LCA is indeed finding solutions of essentially the same quality as the comparison digital algorithms, both in terms of signal recovery of the compressively sensed signal, and in terms of the optimization objective function. When the LCA and digital solutions are compared directly, we find that the average difference in the solutions differs only by a relative mean-squared distance (calculated by $\left\|\widehat{a}_{LCA} - \widehat{a}_{DIG}\right\|_2^2 / \left\|\widehat{a}_{DIG}\right\|_2^2$) of $1.97 \cdot 10^{-4}$ when compared to l1-ls and $6.64 \cdot 10^{-4}$ when compared to GPSR. For comparison, the rMSE of the difference between the l1-ls solutions and the GPSR solutions is $9.71 \cdot 10^{-4}$, meaning that the LCA solutions have variability comparable to what the pair of comparison digital algorithms has between their solutions. We note that the solution differences are significantly larger between all of the algorithms in the regimes where CS recovery is difficult and poor solutions are found by all solvers, as demonstrated by the bottom row of plots in Fig. 50.

6.1.3 LCA convergence time

To observe the potential solution times for the LCA in large-scale CS problems, we compare the convergence of the LCA and GPSR on three specific signals in easy, medium and hard CS recovery problems with the same synthetic data as above (corresponding to different values of δ, ρ). Figure 51 shows the convergence of the relative MSE as a function of time for GPSR and the simulated LCA for three example signals. GPSR times are reported

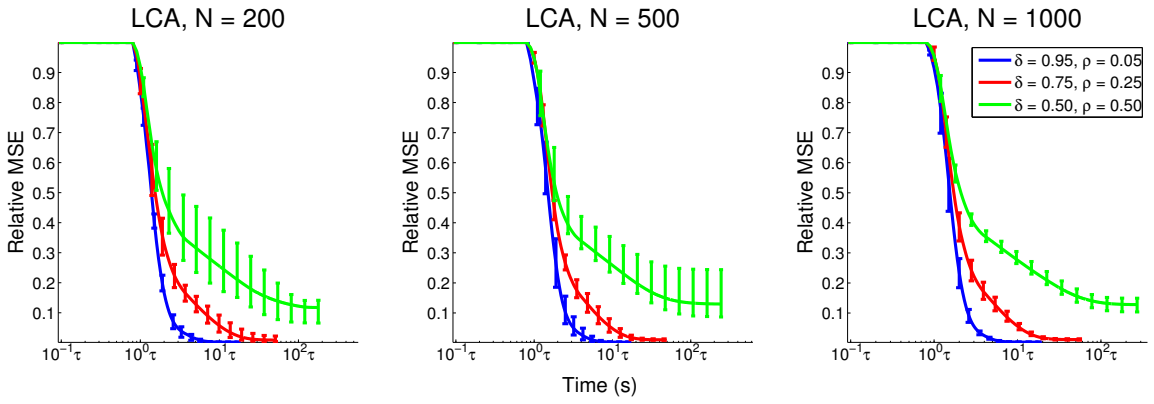

Figure 52: Convergence behavior for the simulated LCA for a number of different problem sizes (N,δ,ρ). Each plot demonstrates the change in convergence based on easy, medium and hard CS recovery problems (i.e., 3 combinations of (δ, ρ)) for $N = 200$ (left), $N = 500$ (middle) and $N = 1000$ (right). While there is no appreciable increase in convergence time with increased problem size (larger N), similar to standard behavior with other optimization algorithms the LCA convergence time does increase with problem difficulty (smaller δ and larger ρ).

using measured CPU[2] time, and LCA times are reported using the number of simulated system time constants τ. The simulation parameters used are identical to the previous simulations. While the solution paths have generally similar characteristics, the time scales are dramatically different. Focusing on the easy and medium CS problems that produce good recovery using ℓ_1 minimization, GPSR is converging in times on the order of 0.3 seconds, whereas the LCA is converging in times on the order of ten time constants (10τ). These simulated times are consistant with the reported times for the hardware implementation described in [27]. We also note that while the results in Fig. 51 are for individual signals for direct comparison with GPSR, the analysis of average case convergence for the LCA shown in Fig. 52 and discussed below also support the same basic conclusions about the LCA convergence time.

Though the time constant of an analog circuit depends on many factors (including the

[2]Time is measured on a Dell Precision Desktop with dual Intel Xeon E5420 Processors and 14GB of DDR3 RAM.

bias current and resulting power consumption of the circuit), $\tau = 10^{-6}$ is a reasonable projected value for a dedicated implementation based on the discussion in [27] and previous reports [205]. Under this assumption, the simulated LCA is converging for CS recovery problems in approximately 10μs of simulated time. Even state of the art digital solutions using high performance computing (either multi-core processing [206] or graphical processing units [207]) currently only achieve speeds in the tens of milliseconds for comparably sized problems. This type of solution speed from the LCA is several orders of magnitude faster than GPSR and could support solvers running in real time at rates of 100 kHz.

Finally, we also investigate the effect of problem size N and problem difficulty (δ, ρ) on the convergence speed of the LCA. For the same parameters corresponding to easy, medium and difficult CS recovery problems as used above, we sample 10 signals at three different problems sizes ($N = 200$, $N = 500$ and $N = 1000$) to perform CS recovery. Figure 52 displays the relative distance of the signal estimate $a^{(t)}$ from the true solution a as a function of simulated time, $||a^{(t)} - a||_2/||a||_2$. The plots are again shown as a function of the simulated time in terms of the number of system time constants τ. As expected, convergence is faster and more reliable (i.e., less variance) for easier recovery problems (i.e., lower sparsity or more measurements). Interestingly, we note that increasing the signal size N does *not* appear to increase the number of time constants required for the LCA solution. In a digital algorithm such as GPSR, while the number of iterations may not increase substantially, the solution time scales with N^2 because the cost of each iteration (e.g., a matrix multiplication) increases significantly. In an analog system like the LCA, increasing the size of a matrix multiply requires increasing the circuit size and complexity, which may increase the time constant [27].

6.1.4 MRI Reconstruction

The previous subsection demonstrated that for stylized problems with synthetic data the LCA can achieve BPDN solutions and signal recoveries comparable to standard digital

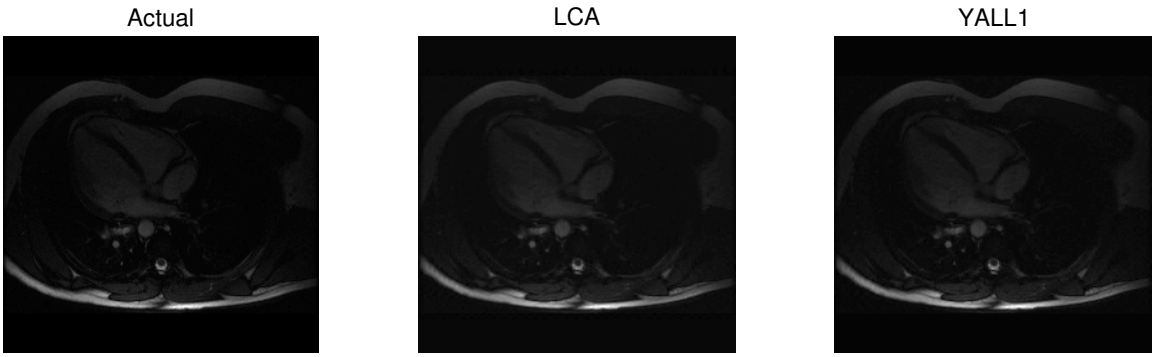

Figure 53: Reconstruction of 256x192 pixel MR images from simulated CS acquisition. The simulated LCA and the comparison digital algorithm (YALL1) find solutions of approximately the same quality in terms of relative MSE and image quality. YALL1 finds the solution in approximately 10s, while the LCA finds the solution in approximately 20 time constants ($20\mu s$ with reasonable estimates of the time constant).

solvers. Furthermore the LCA appears to converge to solutions at speeds that would represent an improvement of several orders of magnitude over digital algorithms. In this section we demonstrate the potential value of this system on a medical imaging application that could be significantly impacted by having real-time CS recovery techniques. Specifically, in this section we simulate the LCA recovery of undersampled MR images to evaluate the solution quality and speed. Compressive MRI is of particular interest because it allows shorter scan times, which improves both patient throughput and lowers risk (e.g., shorter scan times mean that pediatric MRIs may be taken more often without general anesthesia [208]). Furthermore, compressive MR imaging combined with real-time image reconstruction would potentially allow new medical procedures to be performed using real-time 3-D imaging without using ionizing radiation.

We simulate CS data acquisition on 21 frames of a dynamic cardiac MRI sequence[3] by subsampling the Fourier transform of each image (i.e., taking random columns of k-space). Each image is 256x192 pixels, and we recover the images by solving BPDN to find sparse coefficients in a wavelet transform. Specifically, we solve the BPDN optimization

[3]The MRI data used was acquired using a GE 1.5T TwinSpeed scanner (R12M4) using an 8 element cardiac coil.

program where the sensing matrix $\Phi = FW^H$ is an inverse wavelet transform followed by a subsampled Fourier matrix, and recover the image by taking the wavelet transform of the solution to the BPDN problem. The choice of wavelet transforms in this case is very important, as transforms which are coherent with the Fourier subsampling scheme can result in poor results. We follow the work of [208] and use a 4 level 2-dimensional Daubechies wavelet transform as the sparsifying basis. The resulting optimization is more difficult than the synthetic data in the previous two sections because the signals are larger and the images are sparse in a wavelet basis instead of the canonical basis.

We compare results of recovery using the simulated LCA and another standard digital solver YALL1 [193]. Figure 53 shows an example MRI image and its reconstruction using both the LCA and YALL1. The average relative MSE (using $\lambda = 0.001$) over all 21 recovered images was 0.0109 for YALL1 and 0.0106 for the simulated LCA. The relative differences between the LCA and YALL1 solutions was 0.0042, indicating that the solution quality is essentially the same for both approaches. YALL1 took approximately 10 second of computation time to reach this solution (on the same computer platform used in the previous simulations), while the LCA took approximately 20τ simulated seconds. Again using time constant estimates of $\tau = 10^{-6}$, this translates to solution times of 20μs and datarates of approximately 50 kHz. Recovery for such large-scale problems may require more nodes than a single chip can provide. In these cases stringing together a series of smaller chips or developing a block-wise method of recovery would still allow the benefits of using analog hardware for the CS recovery.

6.2 Alternate inference problems in the LCA architecture

Using the basic relationships described in (76) and (77), a variety of cost functions can be optimized in the same basic LCA structure by analytically determining the corresponding activation function.[4] These optimization programs include approximate ℓ_p norms, modified

[4]We also note that a cost function might be easily implementable even in the absence of an analytic formula for the activation function simply by using numerical integration to find a solution and fitting the

ℓ_p norms that attempt to achieve better statistical properties than BPDN, the group/block ℓ_1 norm that induces co-activation structure on the non-zero coefficients, re-weighted ℓ_1 and ℓ_2 algorithms that represent hierarchical statistical models on the coefficients, and classic Tikhonov regularization.

Before exploring specific alternate cost functions in the remainder of this section, it is worthwhile to make a technical note regarding the optimization programs that are possible to implement in the LCA architecture. The strong theoretical convergence guarantees established for the LCA [203] apply to a wide variety of possible systems, but do impose some conditions on the permissible activation functions. We will rely on these same conditions to analytically determine the relationship between the cost and activation functions for the examples in this section. Translated to conditions on the cost functions, the convergence results for the LCA [203] require that the cost functions be positive $\left(\widetilde{C}\left(\boldsymbol{a}\right) \geq 0\right)$, symmetric $\left(\widetilde{C}\left(-\boldsymbol{a}\right) = \widetilde{C}\left(\boldsymbol{a}\right)\right)$, and satisfy the condition that the matrix $\left(\lambda \nabla^2_{\boldsymbol{a}} \widetilde{C}\left(\boldsymbol{a}\right) + \boldsymbol{I}\right)$ is positive definite (i.e., $\lambda \partial^2 C\left(a_k\right) / \partial a_k^2 + 1 > 0$ for separable cost functions). This last condition can intuitively be viewed as requiring that the activation function resulting from (77) has only a single output for a given input.

Some of the cost functions considered here have non-zero derivatives at the origin, leading to a range of values around the origin where $T_\lambda\left(u_k\right)$ is not defined according to the relationship in (76). In these cases, the smallest value for which the threshold function is defined results in a zero-valued output (i.e., $T_\lambda\left(u_k\right) = 0$ at $u_k = \lim_{a_k \to 0^+} \lambda \partial C\left(a_k\right) / \partial a_k$). Since the second derivative condition on the cost function constrains the activation function to be non-decreasing, we can infer that the only allowable value of the activation function must be zero for the regions that are not well-characterized by the relationship in (76). Finally, we note that in most cases we will only consider the behavior of the activation function for $u_k \geq 0$ because the behavior for $u_k < 0$ is implied by the symmetry condition.

resulting curve.

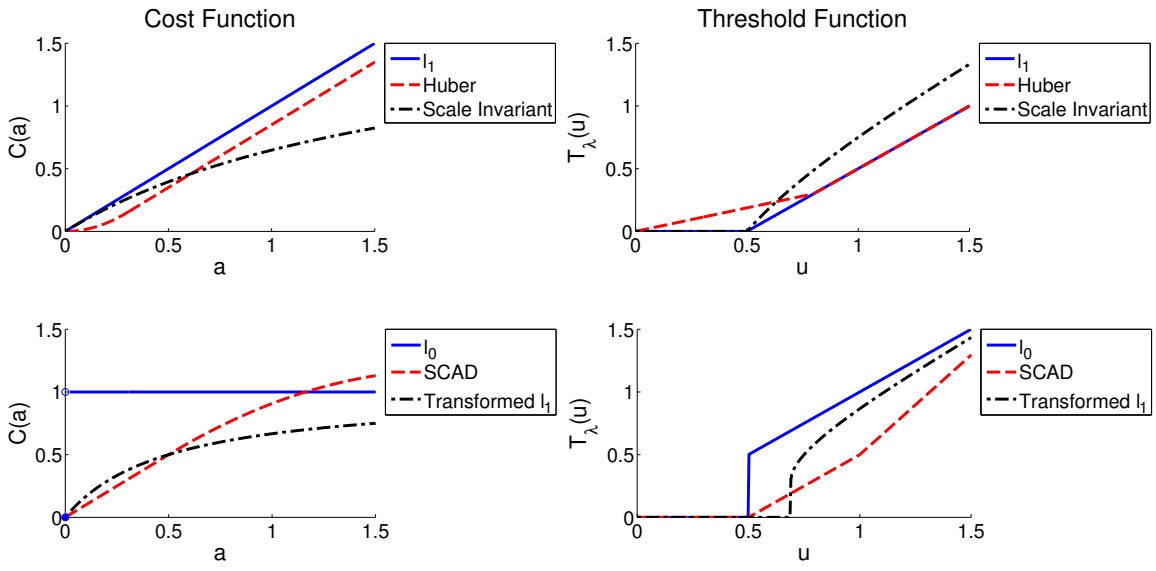

Figure 54: Cost functions and their corresponding thresholding functions. Left: The cost functions are compared for the (top) ℓ_1 with $\lambda = 0.5$, scale invariant Bayes with $\lambda = 0.5$, the Huber cost with $\lambda = 0.5$ and $\epsilon = 0.3$ and (bottom) ℓ_0 with $\lambda = 0.5$, SCAD with $\lambda = 0.5$ and $\kappa = 3.7$ and transformed ℓ_1 with thresh = 0.5 and $\beta = 2$. Right: The corresponding nonlinear activation function which can be used in the LCA to solve the regularized optimization program for each cost function.

6.2.1 Approximate ℓ_p norms $(0 \leq p \leq 2)$

Perhaps the most widely used family of cost functions are the ℓ_p norms $\widetilde{C}(\boldsymbol{a}) = \|\boldsymbol{a}\|_p^p$. These separable cost functions include ideal sparse approximation (i.e., counting non-zeros), BPDN, and Tikhonov Regularization [209] as special cases ($p = 0, 1$ and 2, respectively), and are convex for $p \geq 1$. Furthermore, recent research has shown some benefits of using non-convex ℓ_p norms ($p < 1$) for inverse problems with sparse signal models [210, 211]. While the ideal activation functions can be determined exactly for the three special cases mentioned above ($p = 0, 1$ and 2), it is not possible to analytically determine the activation function for arbitrary values of $0 \leq p \leq 2$. Elad et al. [211] recently introduced several parameterized approximations to the ℓ_p cost functions that are more amenable to analysis. In this section, we use these same approximations to determine activation functions for minimizing approximate ℓ_p norms for $0 \leq p \leq 2$.

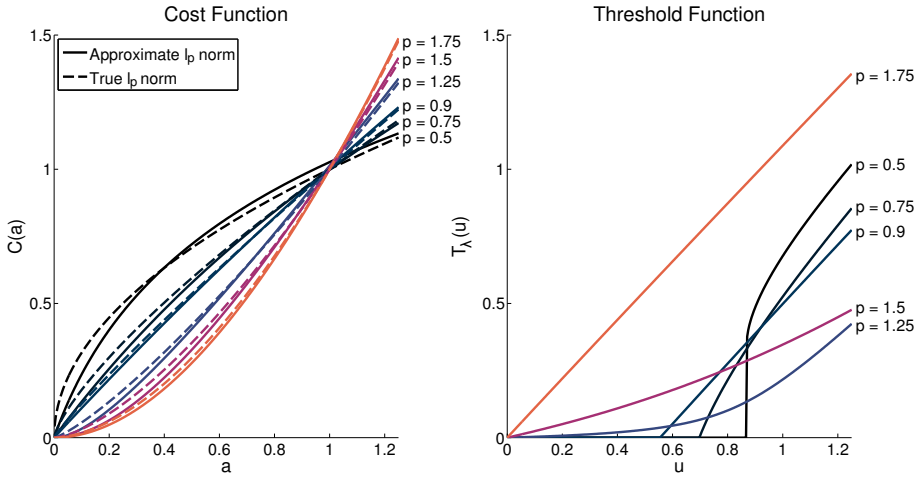

Figure 55: Approximate ℓ_p cost functions and their corresponding thresholding functions. Left: The cost functions are approximated over the parameters c, s for values of p ranging from 0 to 1 (top) and 1 to 2 (bottom). The true ℓ_p costs are shown as dotted lines in the same shades. Using these values of c and s, a nonlinear activation function that can be used in the LCA to solve the optimization is plotted (right) using the thresholding equations for $0 < p < 1$ (top) and $1 < p < 2$ (bottom). The thresholding functions clearly span the ranges between soft and hard thresholding for the lower range of p and between soft thresholding and linear amplification for the upper range of p.

Approximate ℓ_p for $1 \leq p \leq 2$

For $1 \leq p \leq 2$, Elad et al. [211] propose the approximate cost function

$$C(\boldsymbol{a}) = \sum_k \left[c|a_k| - cs \log\left(1 + \frac{|a_k|}{s}\right)\right],$$

as a good match for the true ℓ_p norm for some value of parameters s and c. In the limiting cases, $c = 1$ with $s \to 0$ yields the ℓ_1 norm and $c = 2s$ with $s \to \infty$ yields the ℓ_2 norm. Three intermediate examples for $p = 1.25$, 1.5 and 1.75 are shown in Figure 55. For any specific value of p, we find the best values of c and s by using standard numerical optimization techniques to minimize the squared error to the true cost function over the interval [0,2]. From this cost function, we can differentiate to obtain the relationship between each u_k and a_k as

$$u_k = a_k + \lambda \frac{ca_k}{s + a_k}.$$

We see from this relationship that with $c = 1$ and $s \to 0$, we obtain $a_k = u_k - \lambda$ for $u_k > \lambda$ (i.e., the soft-thresholding function for BPDN), while with $c = 2s$ and $s \to \infty$ we obtain

$a_k = \frac{u_k}{1+2\lambda}$ (i.e., a linear amplifier for Tikhonov Regularization). Solving for a_k in terms of u_k (restricting the solution to be positive and increasing) yields a general relationship for the activation function

$$T_\lambda(u_k) = \frac{1}{2}\left[u_k - s - c\lambda + \sqrt{(u_k - s - c\lambda) + 4u_k s}\right].$$

This solution is shown in Figure 55 for $p = 1.25$, 1.5 and 1.75 for $\lambda = 0.5$.

Approximate ℓ_p for $0 \leq p \leq 1$

For $0 \leq p \leq 1$, Elad et al. [211] also propose the following approximate cost function as a good match for the true ℓ_p norm for some value of parameters s and c:

$$C(a_k) = cs \log\left(1 + \frac{|a_k|}{s}\right),$$

where the parameters $c > 0$ and $s > 0$ can be optimized as above to approximate different values of p. Three approximations for $p = 0.5$, 0.75 and 0.9 are shown in Figure 55. To determine the activation function, we again differentiate and find the appropriate relationship to be

$$a_k + \frac{\lambda cs}{s + a_k} = u_k.$$

Solving for a_k reduces to solving a quadratic equation, which leads to two possible solutions. As above, we restrict the activation function to only include the solution that is positive and increasing, resulting in the activation function

$$T_\lambda(u_k) = \frac{1}{2}\left(u_k - s + \sqrt{(u_k + s)^2 - 4\lambda cs}\right).$$

This activation function is only valid over the range where the output is a positive real number. If $c\lambda \leq s$, this condition reduces to $u_k \geq c\lambda$. More generally, this condition reduces to $u_k \geq 2\sqrt{2cs\lambda} - s$.

6.2.2 Modified ℓ_p norms

While the general ℓ_p norms have historically been very popular cost functions, many people have noted that this approach can have undesirable statistical properties in some instances

(e.g., BPDN can result in biased estimates of large coefficients [212]). To address these issues, many researchers in signal processing and statistics have proposed modified cost functions that attempt to alleviate these statistical concerns. For example, hybrid ℓ_p norms smoothly morph between different norms to capture the most desirable characteristics over different regions. In this section we will demonstrate that many of these modified ℓ_p norms can also be implemented in the basic LCA architecture.

Smoothly Clipped Absolute Deviations

A common goal for modified ℓ_p norms is to retain the continuity of the cost function near the origin demonstrated by the ℓ_1 norm, while using a constant cost function for larger coefficients (similar to the ℓ_0 norm) to avoid statistical biases. One approach to achieving these competing goals is the smoothly clipped absolute deviations (SCAD) penalty [213, 214]. The SCAD approach directly concatenates the ℓ_1 and ℓ_0 norms with a quadratic transition region, resulting in the cost function given by

$$C(a_k) = \begin{cases} a_k & 0 < a_k \leq \lambda \\ \frac{1}{(\kappa-1)\lambda}(a_k\kappa\lambda - \frac{a_k^2}{2} - \frac{\lambda^2}{2}) & \lambda < a_k \leq \kappa\lambda \\ \frac{\lambda}{2}(1+\kappa) & \kappa\lambda < a_k \end{cases},$$

for $\kappa \geq 1$ (κ defines the width of the transition region). An example of this cost function with $\lambda = 0.5$ and $\kappa = 3.7$ is shown in Figure 54.

To obtain the activation function we again solve $\lambda\frac{dC(a_k)}{da_k} + a_k = u_k$ for a_k as a function of u_k. For SCAD (and all of the piecewise cost functions we consider), the activation function can be determined individually for each region, paying careful attention to the ranges of the inputs u_k and outputs a_k to ensure consistency. For $0 < a_k \leq \lambda$, we have $\lambda + a_k = u_k$, implying that $a_k = 0$ for $u_k < \lambda$ and $a_k = u_k - \lambda$ over the interval $\lambda < u_k < 2\lambda$. For $\lambda < a_k \leq \kappa\lambda$, we have

$$\lambda\frac{(\kappa\lambda - a_k)}{(\kappa-1)\lambda} + a_k = u_k \implies a_k = \frac{(\kappa-1)u_k - \kappa\lambda}{\kappa - 2}$$

over the interval $2\lambda < u_k < \kappa\lambda$. Finally, for $\kappa\lambda < a_k$ we have $a_k = u_k$, giving the full activation function

$$a_k = T_\lambda(u_k) = \begin{cases} 0 & u_k \leq \lambda \\ u_k - \lambda & \lambda \leq u_k \leq 2\lambda \\ \frac{\kappa-1}{\kappa-2}u_k - \frac{\kappa\lambda}{\kappa-2} & 2\lambda \leq u_k \leq \kappa\lambda \\ u_k & \kappa\lambda \leq u_k \end{cases},$$

which is shown in Figure 54 for $\lambda = 0.5$ and $\kappa = 3.7$. Note that this activation function requires $\kappa \geq 2$ (Antoniadis and Fan recommend a value of $\kappa = 3.7$ [214]). While this is apparent from consistency arguments once the thresholding function has been derived, this restriction on κ can also be deduced from the condition $\lambda\partial^2 C(a_k)/\partial a_k^2 + 1 > 0$.

Transformed ℓ_1

Similar to the SCAD cost function, the transformed ℓ_1 cost [214, 215] attempts to capture something close to the ℓ_1 norm for small coefficients while reducing the penalty on larger coefficients. Specifically, transformed ℓ_1 uses the fractional cost function given by

$$C(a_k) = \frac{\beta|a_k|}{1+\beta|a_k|},$$

for some $\beta > 0$. An example of this cost with $\beta = 2$ and $\lambda = 0.5$ is shown in Figure 54. After calculating the derivative of the cost function, the activation function can be found by solving

$$\frac{\lambda\beta}{(1+\beta a_k)^2} + a_k = u_k$$

for a_k. Inverting this equation reduces to solving a cubic equation in a_k. The three roots can be calculated analytically, but only one root generates a viable thresholding function

by being both positive and increasing for positive u_k. That root is given by

$$\begin{aligned} a_k &= \frac{\beta u_k - 2}{3\beta} + \frac{2^{\frac{2}{3}}}{6\beta}\left(6\beta u_k - 27\beta^2\lambda + 6\beta^2 u_k^2 + 2\beta^3 u_k^3\right. \\ &+ \left. 3\sqrt{3}\beta^3\sqrt{-\frac{\lambda\left(4\beta^3 u_k^3 + 12\beta^2 u_k^2 - 27\lambda\beta^2 + 12\beta u_k + 4\right)}{\beta^4}} + 2\right)^{\frac{1}{3}} \\ &+ \frac{\beta 2^{\frac{1}{3}}(\beta u_k + 1)^2}{3\left(6\beta u_k - 27\beta^2\lambda + 6\beta^2 u_k^2 + 2\beta^3 u_k^3 + 3\sqrt{3}\beta^3\sqrt{-\frac{\lambda\left(4\beta^3 u_k^3 + 12\beta^2 u_k^2 - 27\lambda\beta^2 + 12\beta u_k + 4\right)}{\beta^4}} + 2\right)^{\frac{1}{3}}} \end{aligned}.$$

This solution is viable only when a_k is real valued, which corresponds to the range $u_k \geq \left(3\left(\frac{\lambda}{4\beta}\right)^{1/3} - \frac{1}{\beta}\right)$. Outside of this range, no viable non-zero solution exists and so $a_k = 0$. The full thresholding function is shown in Figure 54 for $\lambda = 0.5$ and $\beta = 2$..

Huber Function

The Huber cost function [216] aims to modify standard ℓ_2 optimization to improve the robustness to outliers. This cost function consists of a quadratic cost function on smaller values and a smooth transition to an ℓ_1 cost on larger values, given by

$$C(a_k) = \begin{cases} \frac{a_k^2}{2\epsilon} & 0 \leq |a_k| \leq \epsilon \\ |a_k| - \frac{\epsilon}{2} & \epsilon < |a_k| \end{cases}.$$

An example of the Huber cost is shown in Figure 54 for $\lambda = 0.5$ and $\epsilon = 0.3$. As in the case of other piecewise cost functions, we calculate the activation function separately over each interval of interest by calculating the derivative of the cost function in each region. For the first interval, the relationship is given by $\frac{\lambda a_k}{\epsilon} = u_k - a_k$, which obviously gives the activation function $T_\lambda(u_k) = \frac{\epsilon u_k}{\epsilon+\lambda}$ for $|u_k| \leq \epsilon + \lambda$. For the second interval, we have $\lambda\frac{a_k}{|a_k|} = u_k - a_k$, which yields the activation function $T_\lambda(u_k) = u_k\left(1 - \frac{\lambda}{|u_k|}\right)$ for $|u_k| > \epsilon + \lambda$. Putting the pieces together, the full activation function (as expected) is a mixture of the Tikhonov regularization and the soft thresholding used for ℓ_1 optimization given by

$$a_k = T_\lambda(u_k) = \begin{cases} \frac{\epsilon u_k}{\epsilon+\lambda} & |u_k| \leq \epsilon + \lambda \\ u_k\left(1 - \frac{\lambda}{|u_k|}\right) & |u_k| > \epsilon + \lambda \end{cases},$$

which is shown in Figure 54 for $\lambda = 0.5$ and $\epsilon = 0.3$. We can see that as $\epsilon \to 0$, the cost function converges to the ℓ_1 norm and the thresholding function correctly converges back to the soft-threshold function derived earlier using the log-barrier method.

Amplitude Scale Invariant Bayes Estimation

A known problem with using the ℓ_1 norm as a cost function is that it is not scale invariant, meaning that the results can be poor if the amplitude of the input signals changes significantly (assuming a constant value of λ). Many cost functions (including the ones presented above) are heuristically motivated, drawing on intuition and tradeoffs between the behavior of various ℓ_p norms. In contrast, Figueiredo and Nowak [217] approach the problem from the perspective of Bayesian inference with a Jeffreys' prior to determine a cost function with more invariance to amplitude scaling, similar to the non-negative Garrote [218]. We consider here the cost function

$$C(\boldsymbol{a}) = \sum_k -\frac{a_k^2}{4\lambda} + \frac{a_k\sqrt{a_k^2 + 4\lambda^2}}{4\lambda} + \lambda \log\left(a_k + \sqrt{a_k^2 + 4\lambda^2}\right),$$

which is proportional to the one given by Figueiredo and Nowak [217] and is shown in Figure 54 for $\lambda = 0.5$.

Taking the derivative of this cost function, we end up with the relationship between u_k and a_k

$$u_k - a_k = -2\lambda\frac{a_k}{4\lambda} + \frac{2\lambda}{4\lambda}\sqrt{a_k^2 + 4\lambda^2}.$$

Solving for a_k as a function of u_k yields the following activation function,

$$a_k = T_\lambda(u_k) = \begin{cases} 0 & u_k \leq \lambda \\ (u_k^2 - \lambda^2)/u_k & u_k > \lambda \end{cases},$$

matching the results from Figueiredo and Nowak [217]. This activation function is shown in Figure 54 for $\lambda = 0.5$.

6.2.3 Block ℓ_1

While all cost functions discussed earlier in this section have been separable, there is increasing interest in non-separable cost functions that capture structure (i.e., statistical dependencies) between the non-zero coefficients. For example, such structure would be important in performing inference in a complex cell energy model where the energies (i.e., magnitudes) are sparse in a complex-valued signal decomposition (e.g., [219]). Perhaps the most widely cited cost function discussed in this regard is the block ℓ_1 norm (also called the group ℓ_1 norm), which assumes that the coefficients representing $\boldsymbol{x}$ are active in known groups. In this framework, the coefficients are divided into blocks, $\mathcal{A}_l \subset \{a_k\}$ and each block of coefficients $\mathcal{A}_l$ is represented as a vector $\boldsymbol{a}^l$. For our purposes, we assume the blocks are non-overlapping but may have different cardinalities. The block ℓ_1 norm [220] is defined as the ℓ_1 norm over the ℓ_2 norms of the groups,

$$\widetilde{C}(\boldsymbol{a}) = \sum_l \left\|\boldsymbol{a}^l\right\|_2,$$

essentially encouraging sparsity between the blocks (i.e., requiring only a few groups to be active) with no individual penalty on the coefficient values within a block. Because this cost is not separable, the activation function will no longer be a point-wise nonlinearity and will instead have multiple inputs and multiple outputs.

Following the same general approach as above, we calculate the gradient of the cost function for each block,

$$\nabla_{\boldsymbol{a}^l}\widetilde{C}(\boldsymbol{a}) = \frac{\boldsymbol{a}^l}{\left\|\boldsymbol{a}^l\right\|_2},$$

yielding the following relationship between the activation function inputs and outputs

$$\boldsymbol{u}^l = \boldsymbol{a}^l + \lambda\frac{\boldsymbol{a}^l}{\left\|\boldsymbol{a}^l\right\|_2}. \tag{80}$$

While directly solving this relationship for $\boldsymbol{a}^l$ appears difficult, we note that we can simplify the equation by expressing $\left\|\boldsymbol{a}^l\right\|_2$ in terms of $\left\|\boldsymbol{u}^l\right\|_2$. To see this, take the norm of both sides of (80) to get $\left\|\boldsymbol{u}^l\right\|_2 = \left\|\boldsymbol{a}^l\right\|_2 + \lambda$. Substituting back into (80), the relationship simplifies to

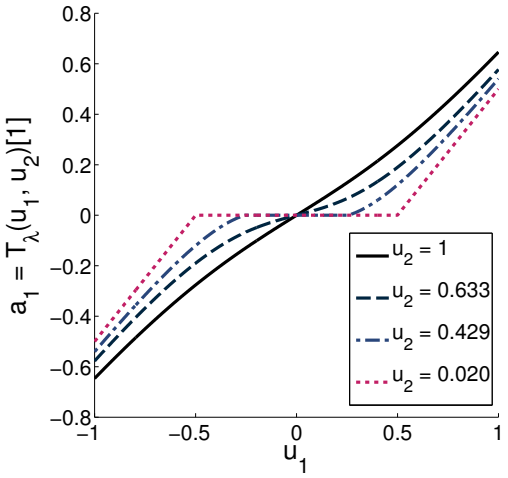

Figure 56: The nonlinear activation function used in the LCA to optimize the non-overlapping group LASSO cost function has multiple inputs and multiple outputs. The plot shows an example thresholding function for both elements in a group of size two ($\lambda = 0.5$), with each line illustrating the nonlinear effect on a_1 while u_2 is held constant.

$$\widetilde{T}_\lambda\left(\boldsymbol{u}^l\right) = \boldsymbol{a}^l = \boldsymbol{u}^l\left(1 - \frac{\lambda}{\left\|\boldsymbol{u}^l\right\|_2}\right)$$

over the range $0 \leq \left\|\boldsymbol{a}^l\right\|_2 = \left\|\boldsymbol{u}^l\right\|_2 - \lambda$, implying $\lambda \leq \left\|\boldsymbol{u}^l\right\|_2$.

This relationship yields the block-wise thresholding function

$$\boldsymbol{a}^l = \widetilde{T}_\lambda\left(\boldsymbol{u}^l\right) = \begin{cases} 0 & \left\|\boldsymbol{u}^l\right\|_2 \leq \lambda \\ \boldsymbol{u}^l\left(1 - \frac{\lambda}{\left\|\boldsymbol{u}^l\right\|_2}\right) & \left\|\boldsymbol{u}^l\right\|_2 > \lambda \end{cases}.$$

This activation function can be thought of as a type of shrinkage operation applied to an entire group of coefficients, with a threshold that depends on the norm of the group inputs. For the case of groups of two elements (with $\lambda = 0.5$), Figure 56 shows the nonlinearities for each of the two states as a function of the value of the other state.

6.2.4 Re-weighted ℓ_1 and ℓ_2

Recent work has also demonstrated that re-weighted ℓ_p norms can achieve better sparsity by iteratively solving a series of tractable convex programs [64,180,181,221]. For example, re-weighted ℓ_1 [180] is an iterative algorithm where a single iteration consists of solving

a weighted ℓ_1 minimization $\left(\widetilde{C}(\boldsymbol{a}) = \sum_k \lambda_k |a_k|\right)$, followed by a weight update according to the rule

$$\lambda_k \propto \frac{1}{|a_k| + \gamma}, \tag{81}$$

where γ is a small parameter. By having λ_k approximately equal to the inverse of the ℓ_1 norm of the coefficient from the previous iteration, this algorithm is more aggressive than BPDN at driving small coefficients to zero and increasing sparsity in the solutions. Similarly, re-weighted ℓ_2 algorithms [181] have also been used to approximate different p-norms with weights updated as

$$\lambda_k \propto \frac{1}{\left(a_k^2 + \gamma\right)^{\left(\frac{p}{2}-1\right)}}.$$

Such schemes have shown many empirical benefits over ℓ_p norm minimization, and recent work on re-weighted ℓ_1 has established theoretical performance guarantees [222] and interpretations as Bayesian inference in a probabilistic model [64].

One of the main drawbacks to re-weighted algorithms in digital architectures is the time required for solving the weighted ℓ_p program multiple times. Of course, it is also not clear that a discrete iterative approach such as this could be mapped to an asynchronous analog computational architecture. Because we have established earlier that the LCA architecture can solve the ℓ_p norm optimizations (and weighted norms are a straightforward extension to those results), it would immediately follow that a dynamical system could be used to perform the optimization necessary for each iteration of the algorithm. While this would be a viable strategy, we show here that even more advantages can be gained by performing the entire re-weighted ℓ_1 algorithm in the context of a dynamical system. Specifically, we consider here a modified version of the LCA where an additional set of dynamics are placed on λ in order to simultaneously optimize the coefficients and coefficient weights in an analog system. While the ideas here are expandable to the general re-weighted case, we focus on results involving the re-weighted ℓ_1 as presented in [64].

The modified LCA is given by the system equations:

$$\begin{aligned} \tau_u \dot{\boldsymbol{u}}(t) &= \Phi^T \boldsymbol{x} - \boldsymbol{u}(t) - \left(\Phi^T \Phi - \boldsymbol{I}\right) \boldsymbol{a}(t) \\ \boldsymbol{a}(t) &= T_\lambda \left(\boldsymbol{u}(t)\right) \\ \tau_\lambda \dot{\lambda}_k(t) &= \lambda_k^{-1}(t) - \nu^{-1} \left(|a_k(t)| + \gamma\right) \end{aligned} \quad .$$

At steady state, $\dot{\boldsymbol{\lambda}} = 0$ which shows that $\lambda_k\ (\infty)$ abides by (81) with ν representing the proportionality constant. While the complete analysis of this expanded analog system is beyond the scope of this paper, we show in Figure 57a simulations which demonstrate that this system reaches a solution of comparable quality to digital iterative methods. Figure 57a plots the relative MSE from a compressed sensing recovery problem with length-1000 vectors from 500 noisy measurements with varying levels of sparsity. We sweep the parameter $\rho = S/M$ from zero to one and set the noise variance to 10^{-4}, with each plot representing the relative MSE averaged over 15 randomly chosen signals. Figure 57(a) plots the recovery quality for three systems: iterative re-weighted ℓ_1 (using GPSR [185] to solve the ℓ_1 iterations), iterative re-weighted ℓ_1 (using the LCA to solve the ℓ_1 iterations), and dynamic re-weighted ℓ_1 which uses the modified LCA described above. It is clear that the three systems are achieving nearly the same quality in their signal recovery. Figure 57b plots the convergence of the recovery as a function of time (in terms of system time constants τ) for the iterative and dynamic re-weighted approaches using the LCA. The dynamically re-weighted system clearly converges more quickly, achieving its final solution in approximately the time it takes to perform two iterations of the traditional re-weighting scheme using the standard LCA.

6.3 Discussion

Sparsity-based signal models have played a significant role in many theories of neural coding across multiple sensory modalities. Despite the interest in the sparse coding hypothesis from the computational and theoretical neuroscience communities, the qualitative nature of much of the supporting evidence leaves significant ambiguity about the ideal form for a

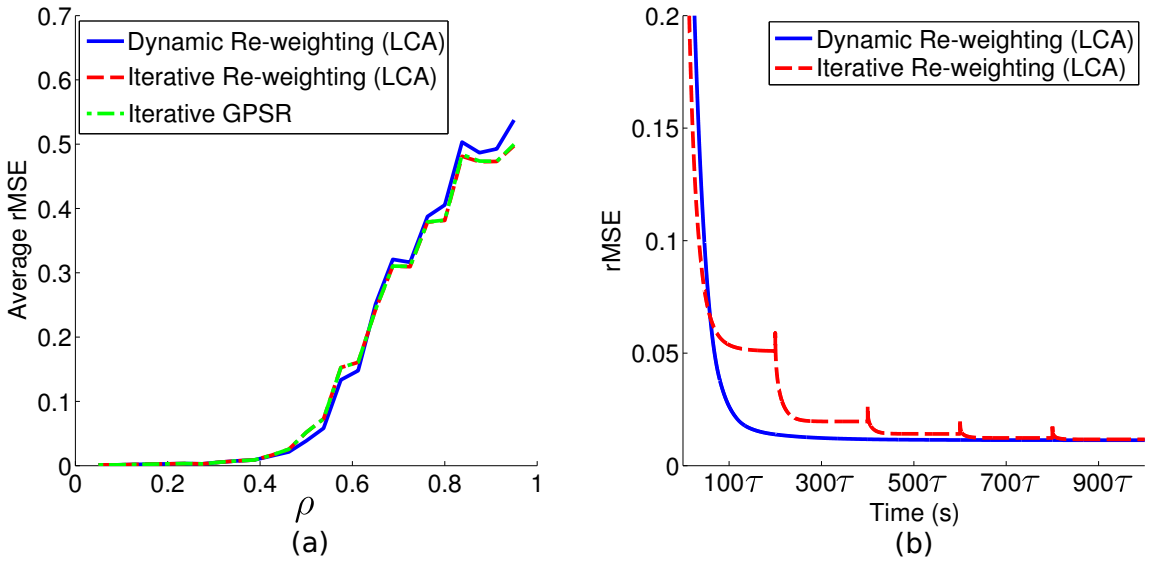

Figure 57: Re-weighted ℓ_1 optimization in digital algorithms and in a modified LCA. (a) Re-weighted ℓ_1 optimization for a signal with $N = 1000$ and $\delta = 0.5$, with ρ swept from 0 to 1. The traditional iterative re-weighting scheme is performed with both a standard digital algorithm (GPSR) and the LCA. For comparison, a dynamic re-weighting scheme where the LCA is modified to have continuous dynamics on the regularization parameter (rather than discrete iterations) is also shown. Each method is clearly achieving similar solutions. (b) The temporal evolution of the recovery relative MSE for a problem with $N = 1000$, $\delta = 0.6$ and $\rho = 0.45$. Solutions are shown for the amount of simulated time (in terms of number of time constants). The dynamically re-weighted system converges in approximately the time it takes to use the LCA to solve two iterations of the traditional re-weighted ℓ_1 algorithm.

sparsity-inducing cost function. While recent trends favor the ℓ_1 norm due the emergence of guarantees in the signal processing literature, there are many sparsity-inducing signal models that may have benefits for neural computation and should be candidate models for neural coding. In this chapter we show that many of the sparsity-inducing cost functions proposed in the signal processing and statistics literatures can be implemented in a single unified dynamical system.

From the results presented here, we conclude that neurally-plausible computational architectures can support a wide variety of sparsity-based signal models, and it is therefore reasonable to consider this broad family of models as reasonable candidates for theories of sensory neural coding. Furthermore, we have shown that even a relatively complex

hierarchical probabilistic model resulting in a re-weighted ℓ_1 inference scheme can be implemented efficiently in a purely analog system. This observation is particularly interesting because it illustrates a fundamental potential advantage of analog computation over digital systems. Specifically, the analog approach to this problem is able to continuously infer two sets of variables jointly, rather than take an iterative approach that fundamentally must wait for the computations in each iteration for one variable to fully converge before inferring the other variable.

Beyond the applicability of these results to theories of neural computation, the recent shift toward optimization as a fundamental computational tool in the modern signal processing toolbox has made it difficult to implement many of these algorithms in applications with significant power constraints or real-time processing requirements. The results of this chapter broaden the scope of problems that could potentially be approached through efficient neuromorphic architectures, both in terms of achievable static decompositions, as well as applications to causal inference of streaming signals. The design and implementation of analog circuits has traditionally been difficult, but recent advances in reconfigurable analog circuits [223] have improved many of the issues related to the design of these systems. In fact, the reconfigurable platform described in [223] has been used to implement a small version of the LCA for solving BPDN [27, 198], and preliminary tests of this implementation are consistent with simulations of the idealized LCA. These results lend encouragement to the idea that efficient analog circuits could be implemented for the variety of cost functions described in this paper.

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

The main goal of this work was to understand how sparse signal structures can be used in conjunction with dynamical systems for efficient signal measurements and inference. In particular, we recall the main questions that we sought to address:

- How can a dynamical system measure sparse signals and can we assess the quality of those measurements?
- How can we recover dynamically or spatially correlated sparse signal from our measurements?
- How can a dynamic system solve sparsity-inducing optimization programs quickly and efficiently?

Overall we find that dynamical systems can, in fact, be highly compatible with sparse signals. For each of these questions, we have provided in this work and the resulting publications (journal papers [1, 2, 10, 11, 20, 21, 26, 27], conference papers [3, 4, 12–15, 22, 28, 224], and conference abstracts [5–9, 16–19, 23–25, 29, 30]) answers by analyzing a range of dynamical systems and signals. Specifically, the three answers are:

A1 In terms of measurement systems, we find that dynamically ESNs can efficiently measure streaming signals. We identify conditions on the inputs that maximize information retention and quantify the information retention via RIP-style conditions which relate the network nodes to the input sequences.

A2 In terms of sparse time-varying signals we derive estimators that make use of both sparsity and dynamic priors. By defining appropriate signal models we can derive both fast, efficient estimators as well as more complex robust estimators. In addition

we show how these results have implications for applications with more general spatially signals by showing the utility of sparsity modeling for hyperspectral imagery.

A3 For optimization implementations, we show that dynamical systems can provide solutions to many important sparsity-inducing optimization programs.

These results are very promising and imply a host of possibilities both for system design and data analysis techniques. Specifically, the our work provides implications for current and future theoretical and algorithmic work, as well as a number of applications.

Theoretical Results and Implications

Our theoretical results show that traditional techniques from the compressive sensing literature can be used to show RIP bounds for dynamically evolving networks as well as convergence guarantees for dynamic filtering of time-varying signals. In the former of these two, we can see that even when a dynamical sensing system can accrue additional correlations in the measurements, quantitative measures, such as the RIP, can still be shown. This implies that the measurements in these systems can still be used to obtain robust and accurate estimates of sparse signals. In the latter results, we can see that, while pessimistic about the actual error bounds, looking at the theoretical convergence rates can help guide parameter choices in dynamic filtering procedures.

For both of these theoretical results, there are many interesting avenues for potential future work. In term of STM for ESNs, our work points to two main theoretical extensions: networks with non-linear nodes and continuous-time networks. Our work here covers the basic case of how ESNs can accrue information over time. While laying the foundation for non-asymptotic analysis of random ESNs, we only consider networks with linear nodes. A number of important network constructions, however, deal with non-linear nodes. Therefore one important extension is understanding the theoretical implications of non-linearities on the ESN's STM. Recent tools from the deep neural network literature [225] which uses techniques from [226, 227] provide one potential avenue to analyze non-linear networks.

While it is not clear how to directly ally these techniques, our success in this work of applying standard compressive sensing tools to dynamical settings gives some precedent that this approach may be viable.

Another important extension of our ESN work is in understanding continuous-time networks. This extension is important in the context of developing compressive sensing systems based on ESNs. While the work on discrete-time inputs here is applicable to recovering band-limited input signals whose Nyquist samples are sparse, continuous-time signals can be much richer in their structure. Specifically, while sparse discrete signals are often defined by their sparsifying dictionary, continuous time models can be defined by more flexible low-dimensional parametrized models (i.e. the signal is described as part of a low-dimensional manifold). As an example, parametrized models can include times and widths of continuous-time shaped pulses. Recent results in recovering low-dimensional parametrized signals can be used along with our work to imply that many additional low-dimensional signal types may be recoverable from random ESNs [228–231].

In terms of our results for dynamic filtering, the main theoretical guarantees focus on convergence of BPDN-DF and RWL1-DF. While we use these guarantees to guide parameter selection for BPDN-DF, the overall bounds for the steady-state estimation errors are rather pessimistic. One main avenue for future work is to improve these bounds for BPDN-DF and to create new bounds for RWL1-DF. Towards these ends, recent work in analyzing optimization programs via statistical dimensions [232], and guarantees for weighted ℓ_1 optimization [233] provide a number of tools which may provide the desired, improved bounds.

Algorithmic Implications

In addition to the theoretical results of this work, we also provide insights into algorithmic development for dynamic and spatial filtering. In particular, we reinforce the idea that propagating the confidence in our prediction through higher order moments is a powerful technique for designing inference procedures for correlated signals. Designing particular

tracking procedures then required finding efficient methods to propagate the confidence variables. In particular, these methods should reflect the actual statistics of the signals being tracked (e.g. variances of Laplacians for sparse signals and covariance matrices for Gaussian signals). This implies that tracking algorithms should be designed from the ground up, rather than the popular approach of modifying the traditional Kalman filtering equations to suit a new need.

As an alternative to designing potentially complex estimation procedures, our work with learning dynamics functions allows us to settle for sub-optimal algorithms, provided the signal models are appropriately learned. Thus a more computationally efficient algorithm can be trained on many data examples to yield improved tracking performance. In future work, we can consider combining the learning procedures with more accurate signal models to continue enhancing sparse signal tracking. For example, we can consider learning the major parameters for the RWL1-DF algorithm in Section 4.5.2.

Implications for Applications

Since many signals and systems have non-trivial dynamic-related correlations, out results have implications for a number of applications. Most generally, out work in tracking sparse signals can be used to further reduce the sampling necessary for systems such as MRI systems. Additionally, adaptations of out tracking work could potentially be used in systems such as RADAR tracking or channel estimation. Our extensions to spatially correlated signals in HSI also have a number of important implications. In particular, the ability to spectrally super-resolve MSI data can yield efficient ways to obtain very high-fidelity remote sensing images. Specifically, MSI typically either has either a much finer spatial resolution or a much larger image area than HSI imagery. Using dictionaries learned from co-located HSI imagery, the MSI images can super-resolved to provide HSI images with either improved spatial resolution or much larger imaging areas.

While more theoretical in nature, our work in STM for ESNs can also have implications for applications. The networked nature of ESNs make them a potential for simplified

models of biological neural networks. Relating our theoretical STM results to psychological experiments on human working memory [234–236] can potentially increase our understanding on how biological networks store information for short-term use.

CHAPTER VIII

APPENDICES

8.1 Bayesian Approach to Kalman Filtering

The Kalman Filtering process seeks to discover an underlying set of state variables $\{\boldsymbol{x}_k\}$ for $k \in [0, n]$ given a set of measurements $\{\boldsymbol{y}_k\}$. The process and measurement equations are both linear and given by

$$\begin{aligned} \boldsymbol{x}_{n+1} &= \boldsymbol{F}_{n+1}\boldsymbol{x}_n + \boldsymbol{\nu}_{o,n+1} \\ \boldsymbol{y}_n &= \boldsymbol{\Phi}\boldsymbol{x}_n + \boldsymbol{\nu}_{d,n}. \end{aligned} \tag{82}$$

The Kalman filter wants to find, at each iteration, the most likely *cause* of the measurement $\boldsymbol{y}_n$ given the approximation made by a flawed estimation (the linear dynamics $\boldsymbol{F}_n$. Figure 58 shows a 2-dimensional graphical depiction. What is important here is not only that we have the measurement and the prediction, but knowledge of *how* each is flawed. In the Kalman case, this knowledge is given by the covariance matrices (essentially fully describing the distribution of the measurement and prediction for the Gaussian case). In Figure 58, this knowledge is represented by the ovals surrounding each point. The power of the Kalman filter comes from it's ability not only to perform this estimation once (a simple Bayesian task), but to use both estimates and knowledge of their distributions to find a distribution for the updated estimate, thus iteratively calculating the best solution for state at each iteration.

While many derivations of the Kalman filter are available, utilizing the orthogonality principle or finding iterative updates to the Best Linear Unbiased Estimator (BLUE), here we derive the Kalman Filter here using a Bayesian approach, where 'best' is interpreted in the Maximum A-Posteriori (MAP) sense instead of an L_2 sense (which for Gaussian

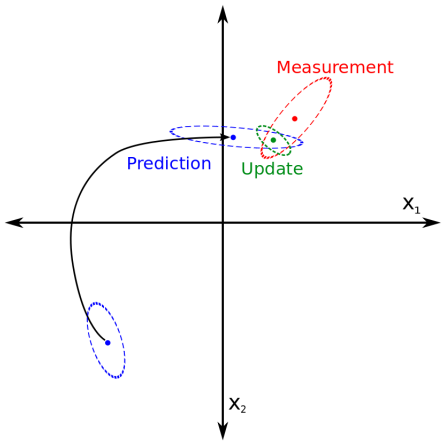

Figure 58: The Kalman filter uses the prediction of a current state based on a previous estimate (blue points) in conjunction with a current measurement (red point) to estimate the true current state (green point). The error in the dynamics (shown here by the blue ovals which represent the covariance)is a combination of the error in the past state and the error in the model of the system. This error in conjunction with the measurement error (the red ovals) allow the covariance of the state update (green oval) to be calculated, propogating forward the confidence of each update.

innovations and measurement noise is the same estimate). Bayesian analysis uses Bayes rule, $p(a|b)p(b) = p(b|a)p(a)$, to express the posterior probability in terms of the likelihood and the prior. In this case we want to optimize over all states $\boldsymbol{x}_k$:

$$\begin{aligned}\{\widehat{\boldsymbol{x}}_k\}_{k\in[0,n]} &= \arg\max\left[\left(\prod_{i=1}^{n} p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1})p(\boldsymbol{y}_i|\boldsymbol{x}_i)\right)p(\boldsymbol{y}_0|\boldsymbol{x}_0)p(\boldsymbol{x}_0)\right] \\ &= \arg\max\left[p(\boldsymbol{y}_n|\boldsymbol{x}_n)p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})\left(\prod_{i=1}^{n-1} p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1})p(\boldsymbol{y}_i|\boldsymbol{x}_i)\right)p(\boldsymbol{y}_0|\boldsymbol{x}_0)p(\boldsymbol{x}_0)\right] \qquad (83)\end{aligned}$$

In order to find a globally optimal solution at the n^{th} time-step only, a marginalization is performed by:

$$\begin{aligned}\widehat{\boldsymbol{x}}_n &= \arg\max_{\boldsymbol{x}_n}\left[\int_{\mathbb{R}^n} p(\boldsymbol{y}_n|\boldsymbol{x}_n)p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})\left(\prod_{i=1}^{n-1} p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1})p(\boldsymbol{y}_i|\boldsymbol{x}_i)\right)p(\boldsymbol{y}_0|\boldsymbol{x}_0)p(\boldsymbol{x}_0)d\{\boldsymbol{x}_l\}_{l\in[0,n-1]}\right] \\ &= \arg\max_{\boldsymbol{x}_n}\left[p(\boldsymbol{y}_n|\boldsymbol{x}_n)\int_{\mathbb{R}^n} p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})\left(\prod_{i=1}^{n-1} p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1})p(\boldsymbol{y}_i|\boldsymbol{x}_i)\right)p(\boldsymbol{y}_0|\boldsymbol{x}_0)p(\boldsymbol{x}_0)d\{\boldsymbol{x}_l\}_{l\in[0,n-1]}\right]\end{aligned}$$

Note that this integral is essentially the prior on $\boldsymbol{x}_n$. Since this prior is an integral of all Gaussian random variables, the result is a Gaussian random variable (Gaussian distributions are self conjugate, and marginalizing over a Gaussian yields a Gaussian). Thus while only

performing a temporally localized update, an updated distribution on $\boldsymbol{x}_n$ is used so that Equation (83) can be written as

$$\widehat{\boldsymbol{x}}_n = \arg\max_{\boldsymbol{x}_n} \left[p(\boldsymbol{y}_n|\boldsymbol{x}_n) p_{\widehat{\boldsymbol{x}}_{n-1}}(\boldsymbol{x}_n) \right]$$

The updated distribution uses all past information to give in essence a likelihood $\boldsymbol{x}_n|\{\boldsymbol{y}_k\}_{k\in[0,n-1]}$. This estimate comes in the form of a probability distribution on the previous estimate $\widehat{\boldsymbol{x}}_{n-1}$, and takes the place of the prior on $\boldsymbol{x}_n$.

The Kalman equations can then be derived by using a MAP estimate. Let the prior on the prediction, $p(x_{n|n-1})$, be determined by Equation (82). In the case of the regular Kalman Filter (a linear process), this is the sum of two multivariate Gaussian distributions. Since the Gaussian is α-stable, this sum is itself a multivariate Gaussian distribution, and can thus be described completely by finding the mean and covariance matrix. The prior on $\widehat{x}_n$ takes the form $\mathcal{N}(\boldsymbol{F}_n\widehat{\boldsymbol{x}}_{n-1}, \boldsymbol{F}_n\boldsymbol{P}_{n-1}\boldsymbol{F}_n^H + \boldsymbol{Q}_n)$. Here $\boldsymbol{P}_{n-1}$ is the correlation matrix of the previous estimate. The MAP estimate is then calculated as:

$$\arg\max_{\widehat{\boldsymbol{x}}_n} p(\widehat{\boldsymbol{x}}_n, \boldsymbol{y}_n) = \arg\max_{\hat{\boldsymbol{x}}_n} p(\boldsymbol{y}_n|\hat{\boldsymbol{x}}_n) p(\hat{\boldsymbol{x}}_n)$$

$$= \arg\max_{\widehat{\boldsymbol{x}}_n} e^{-(\boldsymbol{y}_n - \boldsymbol{\Phi}_n\widehat{\boldsymbol{x}}_n)^H \boldsymbol{R}_n^{-1} (\boldsymbol{y}_n - \boldsymbol{\Phi}_n\hat{\boldsymbol{x}}_n)} e^{-(\hat{\boldsymbol{x}}_n - \boldsymbol{F}_n\hat{\boldsymbol{x}}_{n-1})^H (\boldsymbol{F}_n^H \boldsymbol{P}_{n-1}\boldsymbol{F}_n + \boldsymbol{Q}_n)^{-1} (\hat{\boldsymbol{x}}_n - \boldsymbol{F}_n\hat{\boldsymbol{x}}_{n-1})}$$

$$= \arg\min_{\widehat{\boldsymbol{x}}_n} (\boldsymbol{y}_n - \boldsymbol{\Phi}_n\widehat{\boldsymbol{x}}_n)^H \boldsymbol{R}_n^{-1} (\boldsymbol{y}_n - \boldsymbol{\Phi}_n\hat{\boldsymbol{x}}_n) + (\hat{\boldsymbol{x}}_n - \boldsymbol{F}_n\hat{\boldsymbol{x}}_{n-1})^H (\boldsymbol{F}_n\boldsymbol{P}_{n-1}\boldsymbol{F}_n^H + \boldsymbol{Q}_n)^{-1} (\hat{\boldsymbol{x}}_n - \boldsymbol{F}_n\hat{\boldsymbol{x}}_{n-1})$$

This minimum value can be found analytically by setting the derivative equal to zero:

$$\begin{aligned}
0 &= \frac{\partial}{\partial \widehat{\boldsymbol{x}}_n} \left((\boldsymbol{y}_n - \boldsymbol{\Phi}_n\widehat{\boldsymbol{x}}_n)^H \boldsymbol{R}^{-1} (\boldsymbol{y}_n - \boldsymbol{\Phi}_n\hat{\boldsymbol{x}}_n) + (\hat{\boldsymbol{x}}_n - \boldsymbol{F}_n\hat{\boldsymbol{x}}_{n-1})^H (\boldsymbol{F}_n\boldsymbol{P}_{n-1}\boldsymbol{F}_n^H + \boldsymbol{Q}_n)^{-1} (\hat{\boldsymbol{x}}_n - \boldsymbol{F}_n\hat{\boldsymbol{x}}_{n-1}) \right) \\
&= \frac{\partial}{\partial \widehat{\boldsymbol{x}}_n} (\widehat{\boldsymbol{x}}_n^H (\boldsymbol{\Phi}_n^H \boldsymbol{R}_n^{-1} \boldsymbol{\Phi}_n + (\boldsymbol{F}_n\boldsymbol{P}_{n-1}\boldsymbol{F}_n^H + \boldsymbol{Q}_n)^{-1}) \hat{\boldsymbol{x}}_n - \hat{\boldsymbol{x}}_n^H (\boldsymbol{\Phi}_n^H \boldsymbol{R}_n^{-1} \boldsymbol{y}_n + (\boldsymbol{F}_n\boldsymbol{P}_{n-1}\boldsymbol{F}_n^H)^{-1} \boldsymbol{F}_n\hat{\boldsymbol{x}}_{n-1}) \\
&\qquad - (\boldsymbol{y}_n^H \boldsymbol{R}_n^{-1} \boldsymbol{\Phi}_n + \widehat{\boldsymbol{x}}_{n-1}^H \boldsymbol{F}_n^H (\boldsymbol{F}_n\boldsymbol{P}_{n-1}\boldsymbol{F}_n^H + \boldsymbol{Q}_n)^{-1}) \widehat{\boldsymbol{x}}_n) \\
&= 2(\boldsymbol{\Phi}_n^H \boldsymbol{R}_n^{-1} \boldsymbol{\Phi} + (\boldsymbol{F}_n\boldsymbol{P}_{n-1}\boldsymbol{F}_n^H + \boldsymbol{Q}_n)^{-1}) \widehat{\boldsymbol{x}}_n - 2(\boldsymbol{\Phi}_n^H \boldsymbol{R}_n^{-1} \boldsymbol{y}_n + (\boldsymbol{F}_n\boldsymbol{P}_{n-1}\boldsymbol{F}_n^H + \boldsymbol{Q}_n)^{-1} \boldsymbol{F}_n\widehat{\boldsymbol{x}}_{n-1})
\end{aligned}$$

Let

$$\widehat{\boldsymbol{x}}_{n|n-1} = \boldsymbol{F}_n \widehat{\boldsymbol{x}}_{n-1} \tag{84}$$

and

$$\boldsymbol{P}_{n|n-1} = \boldsymbol{F}_n \boldsymbol{P}_{n-1} \boldsymbol{F}_n^H + \boldsymbol{Q}_n \tag{85}$$

be the projected mean and covariance matrix, respectively:

$$\begin{aligned}
\widehat{\boldsymbol{x}}_n &= \left[\boldsymbol{\Phi}_n^H \boldsymbol{R}_n^{-1} \boldsymbol{\Phi}_n + \boldsymbol{P}_{n|n-1}^{-1}\right]^{-1} \left[\boldsymbol{\Phi}_n^H \boldsymbol{R}_n^{-1} \boldsymbol{y}_n + \boldsymbol{P}_{n|n-1}^{-1} \widehat{\boldsymbol{x}}_{n|n-1}\right] \\
&= \left[\boldsymbol{P}_{n|n-1} - \boldsymbol{P}_{n|n-1}(\boldsymbol{R}_n + \boldsymbol{\Phi}_n \boldsymbol{P}_{n|n-1} \boldsymbol{\Phi}_n^H)^{-1} \boldsymbol{\Phi}_n \boldsymbol{P}_{n|n-1}\right] \left[\boldsymbol{\Phi}_n^H \boldsymbol{R}_n^{-1} \boldsymbol{y}_n + \boldsymbol{P}_{n|n-1}^{-1} \widehat{\boldsymbol{x}}_{n n|n-1}\right] \\
&= \widehat{\boldsymbol{x}}_{n|n-1} - \boldsymbol{K}_n \boldsymbol{\Phi}_n \widehat{\boldsymbol{x}}_{n|n-1} \\
&\qquad + \left[\boldsymbol{P}_{n|n-1} \boldsymbol{\Phi}_n^H \boldsymbol{R}_n^{-1} - \boldsymbol{P}_{n|n-1}(\boldsymbol{R}_n + \boldsymbol{\Phi}_n \boldsymbol{P}_{n|n-1} \boldsymbol{\Phi}_n^H)^{-1} \boldsymbol{\Phi}_n \boldsymbol{P}_{n|n-1} \boldsymbol{\Phi}_n^H \boldsymbol{R}_n^{-1}\right] \boldsymbol{y}_n \\
&= \widehat{\boldsymbol{x}}_{n|n-1} - \boldsymbol{K}_n \boldsymbol{\Phi}_n \widehat{\boldsymbol{x}}_{n|n-1} + \boldsymbol{K}_n \left[(\boldsymbol{\Phi}_n \boldsymbol{P}_{n|n-1} \boldsymbol{\Phi}_n^H + \boldsymbol{R}_n) \boldsymbol{R}_n^{-1} - \boldsymbol{\Phi} \boldsymbol{P}_{n|n-1} \boldsymbol{\Phi}_n^H \boldsymbol{R}_n^{-1}\right] \boldsymbol{y}_n \\
&= \widehat{\boldsymbol{x}}_{n|n-1} - \boldsymbol{K}_n \boldsymbol{\Phi}_n \widehat{\boldsymbol{x}}_{n|n-1} + \boldsymbol{K}_n \boldsymbol{y}_n \\
&= \widehat{\boldsymbol{x}}_{n|n-1} + \boldsymbol{K}_n (\boldsymbol{y}_n - \boldsymbol{\Phi}_n \widehat{\boldsymbol{x}}_{n|n-1})
\end{aligned} \tag{86}$$

Where

$$\boldsymbol{K}_n := \boldsymbol{P}_{n|n-1} \boldsymbol{\Phi}^H \left[\boldsymbol{R}_n + \boldsymbol{\Phi}_n \boldsymbol{P}_{n|n-1} \boldsymbol{\Phi}_n^H\right]^{-1} \tag{87}$$

is the definition of the Kalman gain at time n. This is the exact solution that the Kalman Filter should give as a best estimate of the current state. To continue propagating the estimate to future iterations, the covariance matrix $\boldsymbol{P}_n$ needs to be calculated as well. $\boldsymbol{P}_n$ can then be calculated by simply finding $E[\widehat{\boldsymbol{x}}_{n+1} \widehat{\boldsymbol{x}}_{n+1}^H]$ using the expression derived for the estimate.

$$
\begin{aligned}
E[\widehat{\boldsymbol{x}}_n\widehat{\boldsymbol{x}}_n^H] &= E[(\hat{\boldsymbol{x}}_{n|n-1} + \boldsymbol{K}_n\boldsymbol{y}_n - \boldsymbol{K}_n\boldsymbol{\Phi}_n\hat{\boldsymbol{x}}_{n|n-1})(\hat{\boldsymbol{x}}_{n|n-1} + \boldsymbol{K}_n\boldsymbol{y}_n - \boldsymbol{K}_n\boldsymbol{\Phi}_n\hat{\boldsymbol{x}}_{n|n-1})^H] \\
&= (\boldsymbol{I} - \boldsymbol{K}_n\boldsymbol{\Phi}_n)\boldsymbol{P}_{n|n-1}(\boldsymbol{I} - \boldsymbol{K}_n\boldsymbol{\Phi}_n)^H + \boldsymbol{K}_n\boldsymbol{R}_n\boldsymbol{K}_n^H \\
&= \boldsymbol{P}_{n|n-1} - \boldsymbol{K}_n\boldsymbol{\Phi}\boldsymbol{P}_{n|n-1} - \boldsymbol{P}_{n|n-1}\boldsymbol{\Phi}^H\boldsymbol{K}_n^H + \boldsymbol{K}_n(\boldsymbol{\Phi}_n\boldsymbol{P}_{n|n-1}\boldsymbol{\Phi}_n^H + \boldsymbol{R}_n)\boldsymbol{K}_n^H \\
&= \boldsymbol{P}_{n|n-1} - \boldsymbol{K}_n\boldsymbol{\Phi}_n\boldsymbol{P}_{n|n-1} - \boldsymbol{P}_{n|n-1}\boldsymbol{\Phi}_n^H\boldsymbol{K}_n^H \\
&\qquad + \boldsymbol{P}_{n|n-1}\boldsymbol{\Phi}_n^H\left[\boldsymbol{R}_n + \boldsymbol{\Phi}_n\boldsymbol{P}_{n|n_n}\boldsymbol{\Phi}_n^H\right]^{-1}(\boldsymbol{\Phi}_n\boldsymbol{P}_{n|n-1}\boldsymbol{\Phi}_n^H + \boldsymbol{R}_n)\boldsymbol{K}_n^H \\
&= \boldsymbol{P}_{n|n-1} - \boldsymbol{K}_n\boldsymbol{\Phi}_n\boldsymbol{P}_{n|n-1} - \boldsymbol{P}_{n|n-1}\boldsymbol{\Phi}_n^H\boldsymbol{K}_n^H + \boldsymbol{P}_{n|n_n}\boldsymbol{\Phi}_n^H\boldsymbol{K}_n^H \\
&= \boldsymbol{P}_{n|n-1} - \boldsymbol{K}_n\boldsymbol{\Phi}\boldsymbol{P}_{n|n-1} \qquad (88)
\end{aligned}
$$

The Equations compromising the standard Kalman Filter update are then given by Equations (84), (85), (87), (86), and (88).

8.2 General Temporal Convergence for BPDN-DF

To prove Theorem 6, we first show that the BPDN-DF optimization problem at each iteration is a BPDN problem where the sensing matrix satisfies the RIP with a better constant than the associated inference that does not include dynamic filtering. Theorem 6 is then a direct consequence of using the theoretical guarantees from [125, 237] to obtain a per-iterate error bound, which can be related to the error at the last iteration, allowing for a recursive error bound to be determined. First, we assume that the matrix $\boldsymbol{\Phi}$ satisfies the RIP($2K$, δ) with respect to signals sparse in $\boldsymbol{\Psi}$. We then note that we can combine the first and third terms in the BPDN-DF optimization Equation (41) into an augmented BPDN optimization

$$
\widehat{\boldsymbol{a}}_n = \arg\min_{\boldsymbol{a}} \left\| \begin{bmatrix} \frac{1}{\sqrt{1+\kappa}}\boldsymbol{y} \\ \sqrt{\frac{\kappa}{1+\kappa}} f(\Psi\boldsymbol{a}_{n-1}) \end{bmatrix} - \begin{bmatrix} \frac{1}{\sqrt{1+\kappa}}\boldsymbol{\Phi}\boldsymbol{\Psi} \\ \sqrt{\frac{\kappa}{1+\kappa}}\boldsymbol{\Psi} \end{bmatrix} \boldsymbol{a} \right\|_2^2 + \frac{\gamma}{1+\kappa}\|\boldsymbol{a}\|_1,
$$

Which is essentially trying to solve the BPDN problem with the augmented matrix $\widetilde{\boldsymbol{\Phi}} = [\boldsymbol{\Psi}^T\boldsymbol{\Phi}^T, \sqrt{\kappa}\boldsymbol{\Psi}^T]^T$, and the factor of $1/(1+\kappa)$ is introduced to normalize the columns

of the augmented measurement matrix. Thus the first step is to show that $\widetilde{\mathbf{\Phi}}$ satisfies the RIP as well, and for more favorable constants. Since we assumed that $\mathbf{\Phi}$ had RIP($2K$,δ), we can find the RIP of $\widetilde{\mathbf{\Phi}}$ by observing the upper and lower bounds of the norm of $\|\widetilde{\mathbf{\Phi}}\boldsymbol{a}\|_2^2$ for any $2K$-sparse $\boldsymbol{a}$:

$$\begin{aligned}
\|\widetilde{\mathbf{\Phi}}\mathbf{\Psi}\boldsymbol{a}\|_2^2 &= \frac{1}{1+\kappa}\|\mathbf{\Phi}\mathbf{\Psi}\boldsymbol{a}\|_2^2 + \frac{\kappa}{1+\kappa}\|\boldsymbol{a}\|_2^2 \\
&\leq \frac{C}{1+\kappa}(1+\delta)\|\boldsymbol{a}\|_2^2 + \frac{\kappa}{1+\kappa}\|\boldsymbol{a}\|_2^2 \\
&\leq \frac{C+\kappa+C\delta}{1+\kappa}\|\boldsymbol{a}\|_2^2 \\
&\leq \frac{C+\kappa}{1+\kappa}\left(1+\frac{C}{C+\kappa}\delta\right)\|\boldsymbol{a}\|_2^2,
\end{aligned}$$

similarly, for the lower bound, we obtain $\|\widetilde{\mathbf{\Phi}}\mathbf{\Psi}\boldsymbol{a}\|_2^2 \geq \frac{C+\kappa}{1+\kappa}(1-\frac{C}{C+\kappa}\delta)\|\boldsymbol{a}\|_2^2$. Thus the RIP constants for $\widetilde{\mathbf{\Phi}}$ are $\widetilde{C} = (C+\kappa)/(1+\kappa)$ and $\widetilde{\delta} = C\delta/(C+\kappa)$. Assuming $\mathbf{\Phi}$ is well normalized (i.e. $C = 1$), these expressions reduce to $\widetilde{C} = 1$ and $\widetilde{\delta} = \delta/(1+\kappa)$. Since κ is always positive, this implies that $\widetilde{\delta} < \delta$ and the conditioning on the augmented matrix is improved with respect to the original system. It remains, however, to show that the improved conditioning yields any tangible benefits given that new errors are introduced in the innovations term.

In the BPDN bounds we need to know the ℓ_2 error of the measurements σ_n, which in this case depends on both the actual measurement error as well as the dynamics error. The augmented system has to account for the errors not only in the dynamics model (the innovations term), but also in the previous estimate. We can thus bound the error by observing that

$$\sqrt{\frac{\kappa}{1+\kappa}}(f(\mathbf{\Psi}\widehat{\boldsymbol{a}}_{n-1}) - \mathbf{\Psi}\boldsymbol{a}_n) = \sqrt{\frac{\kappa}{1+\kappa}}(f(\mathbf{\Psi}\widehat{\boldsymbol{a}}_{n-1}) - f(\mathbf{\Psi}\boldsymbol{a}_{n-1}) - \boldsymbol{\nu}_n),$$

Using the smoothness assumption on $f(\cdot)$, we can see that

$$\left\| \sqrt{\frac{\kappa}{1+\kappa}}(f(\mathbf{\Psi}\widehat{\boldsymbol{a}}_{n-1}) - \mathbf{\Psi}\boldsymbol{a}_n) \right\|_2 \leq \sqrt{\frac{\kappa}{1+\kappa}}(f^*\|\boldsymbol{e}+n-1\|_2 + \|\boldsymbol{\nu}_n\|_2),$$

With this inequality, and the assumptions that $\|\boldsymbol{\epsilon}_n\|_2 \leq \bar{\epsilon}$ and $\|\boldsymbol{\nu}_n\|_2 \leq \bar{\nu}$ for all n, the

effective measurement error on the augmented is then

$$\begin{aligned}\sigma_n &= \|\widetilde{\boldsymbol{y}}_n - \widetilde{\boldsymbol{\Phi}}\boldsymbol{a}_n\|_2 \\ &\leq \frac{\|\boldsymbol{\epsilon}_n\|_2}{\sqrt{1+\kappa}} + \sqrt{\frac{\kappa}{1+\kappa}} f^* \|\boldsymbol{e}_{n-1}\|_2 + \sqrt{\frac{\kappa}{1+\kappa}} \|\boldsymbol{\nu}_n\|_2 \\ &\leq \frac{1}{\sqrt{1+\kappa}} \overline{\epsilon} + \sqrt{\frac{\kappa}{1+\kappa}} f^* \|\boldsymbol{e}_{n-1}\|_2 + \sqrt{\frac{\kappa}{1+\kappa}} \overline{\nu} \end{aligned} \tag{89}$$

where f^* is the Lipshitz constant for the function f.

The general form of the BPDN solution satisfies

$$\|\boldsymbol{a}_n - \widehat{\boldsymbol{a}}_n\|_2 \leq C_1 \sigma_n + C_2 \gamma \sqrt{q},$$

where C_1 and C_2 are constants, which can vary depending on the techniques used [125,237]. We can use this bound with the per-time-step σ_n from Equation (89) to find the time-dependent bound

$$\begin{aligned}\|\boldsymbol{e}_n\|_2 &\leq \frac{C_1\left(\overline{\epsilon} + \sqrt{\kappa} f^* \|\boldsymbol{e}_{n-1}\|_2 + \sqrt{\kappa}\overline{\nu}\right)}{\sqrt{1+\kappa}} + \frac{C_2\gamma}{1+\kappa}\sqrt{q} \\ &= \left(C_1 \sqrt{\frac{\kappa}{1+\kappa}} f^*\right) \|\boldsymbol{e}_{n-1}\|_2 + \left(\frac{C_1\overline{\epsilon}}{\sqrt{1+\kappa}} + C_1\sqrt{\frac{\kappa}{1+\kappa}}\overline{\nu} + \frac{C_2\gamma}{1+\kappa}\sqrt{q}\right) \\ &= \beta\|\boldsymbol{e}_{n-1}\|_2 + \alpha \end{aligned}$$

where

$$\beta = C_1 \sqrt{\frac{\kappa}{1+\kappa}} f^*$$

and

$$\alpha = C_1 \frac{1}{\sqrt{1+\kappa}} \overline{\epsilon} + C_1 \sqrt{\frac{\kappa}{1+\kappa}} \overline{\nu} + C_2 \frac{\gamma}{1+\kappa}\sqrt{q}$$

This relationship is essentially a simple linear difference equation and is easily solved for the error at each time step:

$$\|\boldsymbol{e}_n\|_2 \leq \beta^n \left(\|\boldsymbol{e}_0\|_2 - \frac{\alpha}{1-\beta}\right) + \frac{\alpha}{1-\beta}$$

indicating that this algorithm converges linearly with rate β when $\beta < 1$ and the steady state error as $n \to \infty$ is $\|\boldsymbol{e}\infty\|_2 \leq \alpha/(1-\beta)$.

8.3 ISTA-based Temporal convergence for BPDN-DF

To prove Theorem 6, we first need to prove the following theorem about the norm of ISTA's intermediate variables. For ease of notation, we omit the temporal subscripts, assuming that all variables, unless otherwise stated, have temporal subscript n. Additionally, we define the previous (steady state) estimate as $\widehat{\boldsymbol{a}}_n = \widetilde{\boldsymbol{a}}$ and for clarity we refer to $\boldsymbol{a}_n^\dagger$ as the true coefficients at time n. To prove this bound we define two subsets of the sparse vector $\boldsymbol{a}$: J and J'.We define the index subset $J = J[l+1]$ as the union of the current set of active coefficients in the ISTA algorithm $\Gamma[l]$, the q largest elements of the vector $\boldsymbol{u}$ $\Delta[l]$, and the true active set $\Gamma_\dagger$. In [125] it is shown that $|J| = |\Delta[l+1] \cup \Gamma[l] \cup \Gamma_\dagger| \leq S + 2q$. Similarly we define $J` = J[l+2] = \Delta[l+1] \cup \Gamma[l+1] \cup \Gamma_\dagger$.

To start, we bound the energy of $\boldsymbol{u}$ at each algorithmic step at time n with the following lemma:

Lemma 1. *Suppose that the same conditions as in Theorem 6 hold. Additionally, assume that* $\left\|\boldsymbol{a}_n^\dagger\right\|_2^2 \leq b$ *for all n. The vector* $\boldsymbol{u}_J^l$ *(the ISTA variables* $\boldsymbol{u}^l$ *restricted to the support subset J) at each algorithmic iteration l obtained via ISTA (iterating Equation* (42)*) with a step size of* μ *satisfies*

$$\|\boldsymbol{u}_J^l\|_2 \leq \left(|\eta - 1| + \frac{\eta\delta}{1+\kappa}\right)\frac{\gamma}{1+\kappa}\sqrt{q} + \left(\eta + \eta\frac{\kappa f^* + \delta}{\kappa + 1}\right)b + \eta\frac{\sqrt{1+\delta}}{1+\kappa}\overline{\epsilon} + \frac{\eta\kappa}{1+\kappa}\overline{\nu},$$

And with the restriction

$$\widetilde{\eta}(\kappa + \kappa f^* + 1 + \delta)b + \widetilde{\eta}\sqrt{1+\delta}\overline{\epsilon} + \widetilde{\eta}\kappa\overline{\nu} \leq \left(1 - \frac{|\widetilde{\eta}(1+\kappa) - 1| + \widetilde{\eta}\delta)}{1+\kappa}\right)\gamma\sqrt{q}$$

then $\left\|\boldsymbol{u}_J^l\right\|_2$ *is simply bounded by*

$$\left\|\boldsymbol{u}_J^l\right\|_2 \leq \gamma\sqrt{q},$$

Proof:

We start by writing the norm using the definition of $\boldsymbol{u}$ in the ISTA algorithm:

$$\left\|\boldsymbol{u}_J^l\right\|_2 = \left\|\boldsymbol{a}_J^l + \frac{\eta}{1+\kappa}\boldsymbol{\Psi}^T\boldsymbol{\Phi}_J^T\left(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{a}^l\right) + \frac{\eta\kappa}{1+\kappa}\boldsymbol{\Psi}_J^T\left(f(\boldsymbol{\Psi}\widetilde{\boldsymbol{a}}) - \boldsymbol{\Psi}\boldsymbol{a}^l\right)\right\|_2.$$

Using the fact that $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{a}^\dagger + \boldsymbol{\epsilon}$,

$$\left\|\boldsymbol{u}_J^l\right\|_2 = \left\|\boldsymbol{a}_J^l + \frac{\eta}{1+\kappa}\boldsymbol{\Psi}^T\boldsymbol{\Phi}_J^T\boldsymbol{\Phi}\boldsymbol{\Psi}\left(\boldsymbol{a}^\dagger - \boldsymbol{a}^l\right) + \frac{\eta}{1+\kappa}\boldsymbol{\Psi}^T\boldsymbol{\Phi}_J^T\boldsymbol{\epsilon} + \frac{\eta\kappa}{1+\kappa}\boldsymbol{\Psi}_J^T\left(f(\boldsymbol{\Psi}\widetilde{\boldsymbol{a}}) - \boldsymbol{\Psi}\boldsymbol{a}^l\right)\right\|_2 \tag{90}$$

To properly reduce the portion of this expression depending on the dynamics $f(\cdot)$, we note that since the dynamics satisfies $\boldsymbol{\Psi}\boldsymbol{a}_n^\dagger = f(\boldsymbol{\Psi}\boldsymbol{a}_{n-1}^\dagger) + \boldsymbol{\nu}_n$,

$$\begin{aligned} f(\boldsymbol{\Psi}\widetilde{\boldsymbol{a}}) - \boldsymbol{\Psi}\boldsymbol{a}^l &= f(\boldsymbol{\Psi}\widetilde{\boldsymbol{a}}) - \boldsymbol{\Psi}\boldsymbol{a}^\dagger + \boldsymbol{\Psi}\boldsymbol{a}^\dagger - \boldsymbol{\Psi}\boldsymbol{a}^l \\ &= f(\boldsymbol{\Psi}\widetilde{\boldsymbol{a}}) - f(\boldsymbol{\Psi}\boldsymbol{a}_{n-1}^\dagger) - \boldsymbol{\nu} + \boldsymbol{\Psi}\boldsymbol{a}^\dagger - \boldsymbol{\Psi}\boldsymbol{a}^l. \end{aligned}$$

Setting $\widetilde{\eta} = \eta/(1+\kappa)$ and collecting similar terms,

$$\begin{aligned} \left\|\boldsymbol{u}_J^l\right\|_2 &= \left\|\boldsymbol{a}_J^l + \widetilde{\eta}\boldsymbol{\Psi}^T\left(\boldsymbol{\Phi}_J^T\boldsymbol{\Phi} + \kappa\boldsymbol{I}_J\right)\boldsymbol{\Psi}\left(\boldsymbol{a}^\dagger - \boldsymbol{a}^l\right) + \widetilde{\eta}\boldsymbol{\Psi}^T\left(\boldsymbol{\Phi}_J^T\boldsymbol{\epsilon} - \kappa\boldsymbol{\nu}\right) + \widetilde{\eta}\kappa\boldsymbol{\Psi}_J^T\left(f(\boldsymbol{\Psi}\widetilde{\boldsymbol{a}}) - f(\boldsymbol{\Psi}\boldsymbol{a}_{n-1}^\dagger)\right)\right\|_2 \\ &\leq \left\|\boldsymbol{a}_J^l + \widetilde{\eta}\boldsymbol{\Psi}^T(\boldsymbol{\Phi}_J^T\boldsymbol{\Phi} + \kappa\boldsymbol{I}_J)\boldsymbol{\Psi}\left(\boldsymbol{a}^\dagger - \boldsymbol{a}^l\right)\right\|_2 + \widetilde{\eta}\sqrt{1+\delta}\left\|\boldsymbol{\epsilon}\right\|_2 + \widetilde{\eta}\kappa\left\|\boldsymbol{\nu}\right\|_2 \\ &\quad + \widetilde{\eta}\kappa\left\|\boldsymbol{\Psi}_J^T\left(f(\boldsymbol{\Psi}\widetilde{\boldsymbol{a}}) - f(\boldsymbol{\Psi}\boldsymbol{a}_{n-1}^\dagger)\right)\right\|_2 \\ &\leq \left\|\boldsymbol{a}_J^l + \widetilde{\eta}\boldsymbol{\Psi}^T(\boldsymbol{\Phi}_J^T\boldsymbol{\Phi} + \kappa\boldsymbol{I}_J)\boldsymbol{\Psi}\left(\boldsymbol{a}^\dagger - \boldsymbol{a}^l\right)\right\|_2 + \widetilde{\eta}\sqrt{1+\delta}\left\|\boldsymbol{\epsilon}\right\|_2 + \widetilde{\eta}\kappa\left\|\boldsymbol{\nu}\right\|_2 + \widetilde{\eta}\kappa f^*\left\|\boldsymbol{e}_{n-1}\right\|_2 \\ &\leq \left\|(\widetilde{\eta}\boldsymbol{\Phi}_J^T\boldsymbol{\Phi} - (1-\kappa\widetilde{\eta})\boldsymbol{I}_J)\boldsymbol{\Psi}\boldsymbol{a}_J^l\right\|_2 + \left\|\widetilde{\eta}(\boldsymbol{\Phi}_J^T\boldsymbol{\Phi} + \kappa\boldsymbol{I}_J)\boldsymbol{\Psi}\boldsymbol{a}^\dagger\right\|_2 + \widetilde{\eta}\sqrt{1+\delta}\left\|\boldsymbol{\epsilon}\right\|_2 \\ &\quad + \widetilde{\eta}\kappa\left\|\boldsymbol{\nu}\right\|_2 + \widetilde{\eta}\kappa f^*\left\|\boldsymbol{e}_{n-1}\right\|_2. \end{aligned}$$

where the first and third inequalities follow from the triangle inequality, the fact that $\|\boldsymbol{\Psi}\| \leq 1$ and the RIP of $\boldsymbol{\Phi}$, the second inequality follows from the smoothness condition on $f(\cdot)$.

To further simplify the above expression, we use the following inequality which follows from $\boldsymbol{\Phi}$ satisfying RIP($(|J|, \delta)$ with respect to the bases $\boldsymbol{\Psi}$,

$$\left\|\alpha(\boldsymbol{\Psi}\boldsymbol{\Phi})_J^T(\boldsymbol{\Phi}\boldsymbol{\Psi})_J + \beta\boldsymbol{I}_J\right\|_2 \leq |\alpha+\beta| + \alpha\delta. \tag{91}$$

for any constants α, β. Using this inequality, the first two terms of the previous bound can be bounded using the Cauchy-Schwartz inequality and Equation (91)

$$\left\|\boldsymbol{u}_J^l\right\|_2 \leq (|\widetilde{\eta} + \kappa\widetilde{\eta} - 1| + \widetilde{\eta}\delta)\left\|\boldsymbol{a}^l\right\|_2 + \widetilde{\eta}(\kappa + 1 + \delta)\left\|\boldsymbol{a}^\dagger\right\|_2 + \widetilde{\eta}\sqrt{1+\delta}\left\|\boldsymbol{\epsilon}\right\|_2 + \widetilde{\eta}\kappa\left\|\boldsymbol{\nu}\right\|_2 + \widetilde{\eta}\kappa f^*\left\|\boldsymbol{e}_{n-1}\right\|_2$$

Simplifying, we obtain

$$\left\|\boldsymbol{u}_J^l\right\|_2 \leq (|\widetilde{\eta}(1+\kappa)-1|+\widetilde{\eta}\delta)\widetilde{\gamma}\sqrt{q}+\widetilde{\eta}(\kappa+1+\delta)b+\widetilde{\eta}\sqrt{1+\delta}\overline{\epsilon}+\widetilde{\eta}\kappa\overline{\nu}+\widetilde{\eta}\kappa f^*\left\|\boldsymbol{e}_{n-1}\right\|_2,$$

where b is the maximum energy of $\boldsymbol{a}^\dagger$ (i.e. $\left\|\boldsymbol{z}^\dagger\right\|_2 \leq b$) and $\widetilde{\gamma}=\gamma/(1+\kappa)$.

Ideally, this Lemma should be independent of the previous estimation error norm $\|\boldsymbol{e}_{n-1}\|_2$ in order for the Lemma to hold for all n. If we initialize the estimate with the zero vector, the first error has $\|\boldsymbol{e}_0\|_2=\left\|\boldsymbol{a}_0^\dagger\right\|_2 \leq b$. Thus, setting $\|\boldsymbol{e}_{n-1}\|_2 < b$ results in the Lemma statement and it will remain to ensure, through choice of algorithmic parameters, that $\|\boldsymbol{e}_n\|_2 \leq b$ in order for this bound to hold for all n.

First Recursion

With Lemma 1, we seek a recursive expression for the estimation error at algorithmic iteration $l+1$,

$$\begin{aligned}
\left\|\boldsymbol{e}^{l+1}\right\|_2 &= \left\|\boldsymbol{a}^{l+1}-\boldsymbol{a}^\dagger\right\|_2 \\
&= \left\|\boldsymbol{a}^{l+1}-\boldsymbol{u}_{J'}^{l+1}+\boldsymbol{u}_{J'}^{l+1}-\boldsymbol{a}^\dagger\right\|_2 \\
&\leq \left\|\boldsymbol{a}^{l+1}-\boldsymbol{u}_{J'}^{l+1}\right\|_2+\left\|\boldsymbol{u}_{J'}^{l+1}-\boldsymbol{a}^\dagger\right\|_2 \\
&\leq \left\|\boldsymbol{u}_{J'}^{l+1}\right\|_2+\left\|\boldsymbol{a}_{J'}-\boldsymbol{a}^\dagger+\widetilde{\eta}\boldsymbol{\Psi}^T\left(\boldsymbol{\Phi}_J^T\boldsymbol{\Phi}\boldsymbol{\Psi}(\boldsymbol{a}^\dagger-\boldsymbol{a}_{J'}^l)+\boldsymbol{\Phi}_{J'}\boldsymbol{\epsilon}+\kappa(f(\boldsymbol{\Psi}\widetilde{\boldsymbol{a}})-\boldsymbol{\Psi}\boldsymbol{a}^l)\right)\right\|_2 \\
&= \left\|\boldsymbol{u}_{J'}^{l+1}\right\|_2+\left\|(\widetilde{\eta}\boldsymbol{\Psi}^T\boldsymbol{\Phi}_J^T\boldsymbol{\Phi}\boldsymbol{\Psi}-\boldsymbol{I}_{J'})(\boldsymbol{a}^\dagger-\boldsymbol{a}_{J'}^l)+\widetilde{\eta}\boldsymbol{\Psi}^T\boldsymbol{\Phi}_{J'}\boldsymbol{\epsilon}+\widetilde{\eta}\kappa\boldsymbol{\Psi}^T(f(\boldsymbol{\Psi}\widetilde{\boldsymbol{a}})-\boldsymbol{\Psi}\boldsymbol{a}^l)\right\|_2 \\
&\leq \gamma\sqrt{q}+\widetilde{\eta}\kappa f^*\left\|\boldsymbol{e}_{n-1}\right\|_2+\widetilde{\eta}\sqrt{1+\delta}\overline{\epsilon}+\widetilde{\eta}\kappa\overline{\nu}+(|\widetilde{\eta}+\widetilde{\eta}\kappa-1|+\widetilde{\eta}\delta)\|\boldsymbol{e}^l\|_2
\end{aligned}$$

Where the first inequality follows from the triangle inequality, the second inequality follows from the nature of the thresholding function and the definition of $\boldsymbol{u}$ and the third inequality follows from Lemma 1 and a similar set of steps as used to prove Lemma 1 used on the second term. This recursive formula can be solved for $\left\|\boldsymbol{e}^l\right\|_2$ in terms of all other variables as

$$\begin{aligned}
\|\boldsymbol{e}[l]\|_2 \leq (|\eta-1|+\widetilde{\eta}\delta)^l&\left(\left\|\boldsymbol{e}^0\right\|_2-\frac{\gamma\sqrt{q}+\widetilde{\eta}\kappa f^*\left\|\boldsymbol{e}_{n-1}\right\|_2+\widetilde{\eta}\sqrt{1+\delta}\overline{\epsilon}+\widetilde{\eta}\kappa\overline{\nu}}{1-|\eta-1|-\widetilde{\eta}\delta}\right)+ \\
&\frac{\gamma\sqrt{q}+\widetilde{\eta}\kappa f^*\left\|\boldsymbol{e}_{n-1}\right\|_2+\widetilde{\eta}\sqrt{1+\delta}\overline{\epsilon}+\widetilde{\eta}\kappa\overline{\nu}}{1-|\eta-1|-\widetilde{\eta}\delta},
\end{aligned}$$

which gives an upper bound on the steady-state error of

$$\|\boldsymbol{e}_n\|_2 = \left\|\boldsymbol{e}_n^{\infty}\right\|_2 \leq \frac{\gamma\sqrt{q} + \widetilde{\eta}\kappa f^* \|\boldsymbol{e}_{n-1}\|_2 + \widetilde{\eta}\sqrt{1+\delta}\overline{\epsilon} + \widetilde{\eta}\kappa\overline{\nu}}{1 - |\eta - 1| - \widetilde{\eta}\delta}.$$

Second Recursion

This steady state error of the ISTA algorithm with respect to the algorithmic steps can be treated as a new recursive equation on $\|\boldsymbol{e}_n\|_2$ in terms of the signal time-step n which can be solved for $\|\boldsymbol{e}_n\|_2$

$$\begin{aligned}\|\boldsymbol{e}_n\|_2 \quad \leq \quad & \left(\frac{\widetilde{\eta}\kappa f^*}{1 - |\eta - 1| - \widetilde{\eta}\delta}\right)^n \left(\|\boldsymbol{e}_0\|_2 - \frac{\gamma\sqrt{q} + \widetilde{\eta}\sqrt{1+\delta}\overline{\epsilon} + \widetilde{\eta}\kappa\overline{\nu}}{1 - |\eta - 1| - \widetilde{\eta}\delta - \widetilde{\eta}\kappa f^*}\right) \\ & + \frac{\gamma\sqrt{q} + \widetilde{\eta}\sqrt{1+\delta}\overline{\epsilon} + \widetilde{\eta}\kappa\overline{\nu}}{1 - |\eta - 1| - \widetilde{\eta}\delta - \widetilde{\eta}\kappa f^*},\end{aligned}$$

which yields a bound for the error at every iteration n.

The only remaining task is to ensure that $\|\boldsymbol{e}_n\|_2 \leq b$ for all n, in order for Lemma 1 to hold. If b bounds the error at each n, then

$$\beta^n(\|\boldsymbol{e}_0\|_2 - \frac{\alpha}{1-\beta}) + \frac{\alpha}{1-\beta} \leq \beta^n(b - \frac{\alpha}{1-\beta}) + \frac{\alpha}{1-\beta} \leq b$$

holds. Simplifying,

$$\frac{\alpha(1-\beta^n)}{1-\beta} \leq (1-\beta^n)b \rightarrow \frac{\alpha}{1-\beta} \leq b,$$

or in terms of the model and system parameters,

$$\frac{(1+\kappa)\gamma\sqrt{q} + \eta\sqrt{1+\delta}\overline{\epsilon} + \eta\kappa\overline{\nu}}{(1+\kappa)(1 - |\eta - 1|) - \eta\delta - \eta\kappa f^*} \leq b.$$

Since the only parameters we can change at will are κ and γ, we can interpret this bound as a restriction on κ:

$$\begin{cases} \kappa \;>\; \frac{\gamma\sqrt{q}+\eta\sqrt{1+\delta}\overline{\epsilon}-(1-|\eta-1|-\eta\delta)b}{(1-|\eta-1|-\eta f^*)b-\gamma\sqrt{q}-\eta\overline{\nu}} & \text{if} \quad (1 - |\eta - 1| - \eta f^*)b - \gamma\sqrt{q} - \eta\overline{\nu} > 0 \\ \kappa \;<\; \frac{\gamma\sqrt{q}+\eta\sqrt{1+\delta}\overline{\epsilon}-(1-|\eta-1|-\eta\delta)b}{(1-|\eta-1|-\eta f^*)b-\gamma\sqrt{q}-\eta\overline{\nu}} & \text{if} \quad (1 - |\eta - 1| - \eta f^*)b - \gamma\sqrt{q} - \eta\overline{\nu} < 0 \end{cases}$$

Which completes the proof.

8.4 RIP for Single Input with Optimal Feed-Forward Vectors

In this appendix, we show that the matrix $\boldsymbol{\Phi} = \boldsymbol{U}\widetilde{\boldsymbol{Z}}\boldsymbol{F}$ satisfies the RIP under the conditions stated in Equation (21) of the main text in order to prove Theorem 3.1.3. We note that [53] shows that for the canonical basis ($\boldsymbol{\Psi} = \boldsymbol{I}$), the bounds for M can be tightened to $M \geq \max\left\{C\frac{S}{\delta^2}\log^4 N,\ C'\frac{S}{\delta^2}\log\eta^{-1}\right\}$ using a more complex proof technique than we will employ here. For $\eta = \frac{1}{N}$, the result in [53] represents an improvement of several $\log(N)$ factors when restricted to only the canonical basis for $\boldsymbol{\Psi}$. We also note that the scaling constant C found in the general RIP definition of Equation (5) of the main text is unity due to the $\sqrt{M}$ scaling of $\boldsymbol{z}$.

While the proof of Theorem 3.1.3 is fairly technical, the procedure follows very closely the proof of Theorem 8.1 from [53] on subsampled discrete time Fourier transform (DTFT) matrices. While the basic approach is the same, the novelty in our presentation is the incorporation of the sparsity basis $\boldsymbol{\Psi}$ and considerations for a real-valued connectivity matrix $\boldsymbol{W}$.

Before beginning the proof of this theorem, we note that because $\boldsymbol{U}$ is assumed unitary, $\|\boldsymbol{\Phi\Psi x}\|_2 = \|\widetilde{\boldsymbol{Z}}\boldsymbol{F\Psi x}\|_2$ for any signal $\boldsymbol{x}$. Thus, it suffices to establish the conditioning properties of the matrix $\widehat{\boldsymbol{\Phi}} := \widetilde{\boldsymbol{Z}}\boldsymbol{F\Psi}$. For the upcoming proof, it will be useful to write this matrix as a sum of rank-1 operators. The specific rank-1 operator that will be useful for our purposes is $X_l X_l^H$ with $X_l^H := \boldsymbol{F}_l^H\boldsymbol{\Psi}$, the conjugate of the l-th row of $\boldsymbol{F\Psi}$, where $\boldsymbol{F}_l^H := \left[1,\ e^{jw_l},\ \cdots,\ e^{jw_l(N-1)}\right] \in \mathbb{C}^N$ is the conjugated l-th row of $\boldsymbol{F}$. Because of the way the "frequencies" $\{w_m\}$ are chosen, for any $l > \frac{M}{2}$, $X_l = X_{l-\frac{M}{2}}^*$. The l-th row of $\widehat{\boldsymbol{\Phi}}$ is $\widetilde{z}_l X_l^H$ where $\widetilde{z}_l$ is the l-th diagonal entry of the diagonal matrix $\widetilde{\boldsymbol{Z}}$, meaning that we can use the sum of rank-1 operators to write the decomposition $\widehat{\boldsymbol{\Phi}}^H\widehat{\boldsymbol{\Phi}} = \sum_{l=1}^M |\widetilde{z}_l|^2 X_l X_l^H$. If we define the random variable $\boldsymbol{B} := \widehat{\boldsymbol{\Phi}}^H\widehat{\boldsymbol{\Phi}} - \boldsymbol{I}$ and the norm $\|\boldsymbol{B}\|_S := \sup_{y \text{ is } S\text{-sparse}} \frac{\boldsymbol{y}^H\boldsymbol{B}\boldsymbol{y}}{\boldsymbol{y}^H\boldsymbol{y}}$, we can equivalently say that $\widehat{\boldsymbol{\Phi}}$ has RIP conditioning δ if

$$\|\boldsymbol{B}\|_S := \left\|\widehat{\boldsymbol{\Phi}}^H\widehat{\boldsymbol{\Phi}} - \boldsymbol{I}\right\|_S = \left\|\sum_{l=1}^M |\widetilde{z}_l|^2 X_l X_l^H - \boldsymbol{I}\right\|_S \leq \delta.$$

To aid in the upcoming proof, we make a few preliminary observations and rewrite the quantities of interest in some useful ways. First, because of the correspondences between the summands in $\widehat{\mathbf{\Phi}}^H\widehat{\mathbf{\Phi}}$ (i.e. $X_l = X^*_{l-M/2}$), we can rewrite $\widehat{\mathbf{\Phi}}^H\widehat{\mathbf{\Phi}}$ as

$$\widehat{\mathbf{\Phi}}^H\widehat{\mathbf{\Phi}} \;=\; \sum_{l=1}^{M/2} |\widetilde{z}_l|^2 X_l X_l^H + \sum_{l=1}^{M/2} |\widetilde{z}_l|^2 \left(X_l X_l^H\right)^*,$$

making clear the fact that there are only $\frac{M}{2}$ independent w_m's. Under the assumption of Theorem 3.1.3, $\widetilde{z}_l = \frac{1}{\sqrt{M}}$ for $l = 1, \cdots, M$. Therefore,

$$\mathcal{E}\left[\sum_{l=1}^{M/2} |\widetilde{z}_l|^2 X_l X_l^H\right] = \sum_{l=1}^{M/2} |\widetilde{z}_l|^2 \mathcal{E}\left[X_l X_l^H\right] = \sum_{l=1}^{M/2} \frac{1}{M} \mathbf{\Psi}^H \mathcal{E}\left[\boldsymbol{F}_l \boldsymbol{F}_l^H\right] \mathbf{\Psi} = \frac{1}{2}\boldsymbol{I},$$

where it is straightforward to check that $\mathcal{E}\left[\boldsymbol{F}_l \boldsymbol{F}_l^H\right] = \boldsymbol{I}$. By the same reasoning, we also have $\mathcal{E}\left[\sum_{l=1}^{M/2} |\widetilde{z}_l|^2 \left(X_l X_l^H\right)^*\right] = \frac{1}{2}\boldsymbol{I}$. This implies that we can rewrite $\boldsymbol{B}$ as

$$\begin{aligned}
\boldsymbol{B} &= \sum_{l=1}^{M} \left(|\widetilde{z}_l|^2 X_l X_l^H\right) - \boldsymbol{I} \\
&= \left(\sum_{l=1}^{M/2} |\widetilde{z}_l|^2 X_l X_l^H - \frac{1}{2}\boldsymbol{I}\right) + \left(\sum_{l=1}^{M/2} |\widetilde{z}_l|^2 \left(X_l X_l^H\right)^* - \frac{1}{2}\boldsymbol{I}\right) \\
&=: \boldsymbol{B}_1 + \boldsymbol{B}_2.
\end{aligned}$$

The main proof of the theorem has two main steps. First, we will establish a bound on the moments of the quantity of interest $\|\boldsymbol{B}\|_S$. Next we will use these moments to derive a tail bound on $\|\boldsymbol{B}\|_S$, which will lead directly to the RIP statement we seek. The following two lemmas from the literature will be critical for these two steps.

Lemma 2 (Lemma 8.2 of [53])**.** *Suppose $M \geq S$ and suppose we have a sequence of (fixed) vectors $Y_l \in \mathbb{C}^N$ for $l = 1, \cdots, M$ such that $\kappa := \max_{l=1,\cdots,M} \|Y_l\|_\infty < \infty$. Let $\{\xi_l\}$ be a Rademacher sequence, i.e., a sequence of i.i.d. ± 1 random variables. Then for $p = 1$ and for $p \in \mathbb{R}$ and $p \geq 2$,*

$$\begin{aligned}
\left(\mathcal{E}\left[\left\|\sum_{l=1}^{M} \xi_l Y_l Y_l^H\right\|_S^p\right]\right)^{1/p} \;\leq\; & C' C^{1/p} \kappa \sqrt{p}\sqrt{S}\log(100S)\sqrt{\log(4N)\log(10M)} \\
& \times \sqrt{\left\|\sum_{l=1}^{M} Y_l Y_l^H\right\|_S},
\end{aligned}$$

where C, C' are universal constants.

Lemma 3 (Adapted from Proposition 6.5 of [53])**.** *Suppose Z is a random variable satisfying*

$$(\mathcal{E}[|Z|^p])^{1/p} \leq \alpha\beta^{1/p}p^{1/\gamma},$$

for all $p \in [p_0,\ p_1]$, and for constants $\alpha, \beta, \gamma, p_0, p_1$. Then, for all $u \in [p_0^{1/\gamma},\ p_1^{1/\gamma}]$,

$$\mathcal{P}\left[|Z| \geq e^{1/\gamma}\alpha u\right] \leq \beta e^{-u^\gamma/\gamma}.$$

Armed with this notation and these lemmas, we now prove Theorem 3.1.3:

Proof. We seek to show that under the conditions on M in Theorem 3.1.3, $\mathcal{P}[\|\boldsymbol{B}\|_S > \delta] \leq \eta$. Since $\boldsymbol{B} = \boldsymbol{B}_1 + \boldsymbol{B}_2$ and $\{\|\boldsymbol{B}_1\|_S \leq \delta/2\} \cap \{\|\boldsymbol{B}_2\|_S \leq \delta/2\} \subset \{\|\boldsymbol{B}\|_S \leq \delta\}$, then,

$$\mathcal{P}[\|\boldsymbol{B}\|_S > \delta] \leq \mathcal{P}[\|\boldsymbol{B}_1\|_S > \delta/2] + \mathcal{P}[\|\boldsymbol{B}_2\|_S > \delta/2].$$

Thus, it will suffice to bound $\mathcal{P}[\|\boldsymbol{B}_1\|_S > \delta/2] \leq \eta/2$ since $\boldsymbol{B}_2 = \boldsymbol{B}_1^*$ implies that $\mathcal{P}[\|\boldsymbol{B}_2\|_S > \delta/2] \leq \eta/2$. In this presentation we let C, C' be some universal constant that may not be the same from line to line.

To begin, we use Lemma 2 to bound $E_p := \left(\mathcal{E}\left[\|\boldsymbol{B}_1\|_S^p\right]\right)^{1/p}$ by setting $Y_l = \widetilde{z}_l^* X_l$ for $l = 1, \cdots, \frac{M}{2}$. To meet the conditions of Lemma 2 we use a standard "symmetrization" manipulation (see Lemma 6.7 of [53]). Specifically, we can write:

$$\begin{aligned} E_p &= \left(\mathcal{E}\left[\|\boldsymbol{B}_1\|_S^p\right]\right)^{1/p} \\ &\leq 2\left(\mathcal{E}\left[\left\|\sum_{l=1}^{M/2} \xi_l Y_l Y_l^H\right\|_S^p\right]\right)^{1/p} \\ &= 2\left(\mathcal{E}\left[\left\|\sum_{l=1}^{M/2} \xi_l |\widetilde{z}_l|^2 X_l X_l^H\right\|_S^p\right]\right)^{1/p}, \end{aligned}$$

where now the expectation is over the old random sequence $\{w_l\}$, together with a newly added Rademacher sequence $\{\xi_l\}$. Applying the law of iterated expectation and Lemma 2,

we have for $p \geq 2$:

$$\begin{aligned}
E_p^p &:= \mathcal{E}\left[\|\boldsymbol{B}_1\|_S^p\right] \\
&\leq 2^p \mathcal{E}\left[\mathcal{E}\left[\left\|\sum_{l=1}^{M/2} \xi_l |\widetilde{z}_l|^2 X_l X_l^H\right\|_S^p \mid \{w_l\}\right]\right] \\
&\leq \left(2C'C^{1/p}\sqrt{p}\kappa\sqrt{S}\log(100S)\sqrt{\log(4N)\log(5M)}\right)^p \mathcal{E}\left[\left\|\sum_{l=1}^{M/2} |\widetilde{z}_l|^2 X_l X_l^H\right\|_S^{p/2}\right] \\
&\leq \left(C^{1/p}\sqrt{p}\kappa\sqrt{C'S\log^4(N)}\right)^p \mathcal{E}\left[\left(\left\|\sum_{l=1}^{M/2}\left(|\widetilde{z}_l|^2 X_l X_l^H - \frac{1}{2}\boldsymbol{I}\right)\right\|_S + \frac{1}{2}\|I\|_S\right)^{p/2}\right] \\
&\leq \left(C^{1/p}\sqrt{p}\kappa\sqrt{C'S\log^4(N)}\right)^p \sqrt{\mathcal{E}\left[\left(\left\|\sum_{l=1}^{M/2}\left(|\widetilde{z}_l|^2 X_l X_l^H - \frac{1}{2}\boldsymbol{I}\right)\right\|_S + \frac{1}{2}\right)^p\right]} \\
&\leq \left(C^{1/p}\sqrt{p}\kappa\sqrt{C'S\log^4(N)}\right)^p \sqrt{\left(E_p + \frac{1}{2}\right)^p}.
\end{aligned}$$

In the first line above, the inner expectation is over the Rademacher sequence $\{\xi_l\}$ (where we apply Lemma 2) while the outer expectation is over the $\{w_l\}$. The third line uses the triangle inequality for the $\|\cdot\|_S$ norm, the fourth line uses Jansen's inequality, and the fifth line uses triangle inequality for moments norm (i.e., $(\mathcal{E}[|X+Y|^p])^{1/p} \leq (\mathcal{E}[|X|^p])^{1/p} + (\mathcal{E}[|Y|^p])^{1/p}$). To get to $\log^4 N$ in the third line, we used our assumption that $N \geq M$, $N \geq S$ and $N \geq O(1)$ in Theorem 3.1.3. Now using the definition of κ from Lemma 2, we can bound this quantity as:

$$\kappa := \max_l \|Y_l\|_\infty = \max_l |\widetilde{z}_l| \|X_l\|_\infty = \frac{1}{\sqrt{M}} \max_l \|X_l\|_\infty = \frac{1}{\sqrt{M}} \max_{l,n} |\langle \boldsymbol{F}_l, \boldsymbol{\Psi}_n \rangle| \leq \frac{\mu(\boldsymbol{\Psi})}{\sqrt{M}}.$$

Therefore, we have the following implicit bound on the moments of the random variable of interest

$$E_p \leq C^{1/p}\sqrt{p}\sqrt{\frac{C'S\mu(\boldsymbol{\Psi})^2\log^4(N)}{M}}\sqrt{E_p + \frac{1}{2}}.$$

The above can be written as $E_p \leq a_p\sqrt{E_p + \frac{1}{2}}$, where $a_p = C^{1/p}\sqrt{p}\sqrt{\frac{4C'S\mu(\boldsymbol{\Psi})^2\log^4(N)}{M}}$. By squaring, rearranging the terms and completing the square, we have $E_p \leq \frac{a_p^2}{2} + a_p\sqrt{\frac{1}{2} + \frac{a_p^2}{4}}$.

By assuming $a_p \leq \frac{1}{2}$, this bound can be simplified to $E_p \leq a_p$. Now, this assumption is equivalent to having an upper bound on the range of values of p:

$$\begin{aligned} a_p \leq \frac{1}{2} \quad &\Leftrightarrow \quad \sqrt{p} \leq \frac{1}{2C^{1/p}} \sqrt{\frac{M}{4C'S\mu(\boldsymbol{\Psi})^2 \log^4(N)}} \\ &\Leftrightarrow \quad p \leq \frac{M}{16C^{2/p}C'S\mu(\boldsymbol{\Psi})^2 \log^4(N)}. \end{aligned}$$

Hence, by using Lemma 3 with $\alpha = \sqrt{\frac{C'S\mu(\boldsymbol{\Psi})^2 \log^4(N)}{M}}$, $\beta = C$, $\gamma = 2$, $p_0 = 2$, and $p_1 = \frac{M}{16C^{2/p}C'S\mu(\boldsymbol{\Psi})^2 \log^4(N)}$ we obtain the following tail bound for $u \in [\sqrt{2}, \sqrt{p_1}]$:

$$\mathcal{P}\left[\|\boldsymbol{B}_1\|_S \geq e^{1/2} \sqrt{\frac{C'S\mu(\boldsymbol{\Psi})^2 \log^4(N)}{M}} u\right] \leq Ce^{-u^2/2}.$$

If we pick $\delta < 1$ such that

$$e^{1/2} \sqrt{\frac{C'S\mu(\boldsymbol{\Psi})^2 \log^4(N)}{M}} u \leq \frac{\delta}{2} \tag{92}$$

and u such that

$$Ce^{-u^2/2} \leq \frac{\eta}{2} \quad \Leftrightarrow \quad u \geq \sqrt{2\log(2C\eta^{-1})},$$

then we have our required tail bound of $\mathcal{P}[\|\boldsymbol{B}_1\|_S > \delta] \leq \eta/2$. First, observe that Equation (92) is equivalent to having

$$M \geq \frac{CS\mu(\boldsymbol{\Psi})^2 \log^4(N) \log(\eta^{-1})}{\delta^2}.$$

Also, because of the limited range of values u can take (i.e., $u \in [\sqrt{2}, \sqrt{p_1}]$), we require that

$$\begin{aligned} \sqrt{2\log(2C\eta^{-1})} \quad &\leq \quad \sqrt{\frac{M}{16C^{2/p}C'S\mu(\boldsymbol{\Psi})^2 \log^4(N)}} = \sqrt{p_1} \\ &\Leftrightarrow \quad M \geq CS\mu(\boldsymbol{\Psi})^2 \log^4(N) \log(\eta^{-1}), \end{aligned}$$

which, together with the earlier condition on M, completes the proof.

□

8.5 RIP for Single Input with Gaussian Feed-Forward Vectors

In this appendix we extend the RIP analysis of Appendix 8.4 to the case when $\boldsymbol{z}$ is chosen to be a Gaussian i.i.d. vector, as presented in Theorem 3.1.3. It is unfortunate that with the additional randomness in the feed-forward vector, the same proof procedure as in Theorem 3.1.3 cannot be used. In the proof of Theorem 3.1.3, we showed that the random variable $\|Z_1\|_S$ has p-th moments that scale like $\alpha\beta^{1/p}p^{1/2}$ (through Lemma 2) for a range of p which suggests that it has a sub-gaussian tail (i.e., $\mathcal{P}\left[\|Z_1\|_S > u\right] \leq Ce^{-u^2/2}$) for a range of deviations u. We then used this tail bound to bound the probability that $\|Z_1\|_S$ exceeds a fixed conditioning δ. With Gaussian uncertainties in the feed-forward vector z, Lemma 2 will not yield the required sub-gaussian tail but instead gives us moments estimates that result in sub-optimal scaling of M with respect to N. Therefore, we will instead follow the proof procedure of Theorem 16 from [238] that will yield the better measurement rate given in Theorem 3.1.3.

Let us begin by recalling a few notations from the proof of Theorem 3.1.3 and by introducing further notations that will simplify our exposition later. First, recall that we let X_l^H be the l-th row of $\boldsymbol{F\Psi}$. Thus, the l-th row of our matrix of interest $\widehat{\boldsymbol{\Phi}} = \widetilde{\boldsymbol{Z}}\boldsymbol{F\Psi}$ is $\widetilde{z}_l X_l^H$ where $\widetilde{z}_l$ is the l-th diagonal entry of the diagonal matrix $\widetilde{\boldsymbol{Z}}$. Whereas before, $\widetilde{z}_l = \frac{1}{\sqrt{M}}$ for any $l = 1, \cdots, M$, here it will be a random variable. To understand the resulting distribution of $\widetilde{z}_l$, first note that for the connectivity matrix $\boldsymbol{W}$ to be real, we need to assume that the second $\frac{M}{2}$ columns of $\boldsymbol{U}$ are complex conjugates of the first $\frac{M}{2}$ columns. Thus, we can write $\boldsymbol{U} = [\boldsymbol{U}_R \mid \boldsymbol{U}_R] + j\,[\boldsymbol{U}_I \mid -\boldsymbol{U}_I]$, where $\boldsymbol{U}_R, \boldsymbol{U}_I \in \mathbb{R}^{M\times\frac{M}{2}}$. Because $\boldsymbol{U}^H\boldsymbol{U} = \boldsymbol{I}$, we can deduce that $\boldsymbol{U}_R^T\boldsymbol{U}_I = \boldsymbol{0}$ and that the ℓ_2 norms of the columns of both $\boldsymbol{U}_R$ and $\boldsymbol{U}_I$ are $\frac{1}{\sqrt{2}}$.[1]

With these matrices $\boldsymbol{U}_R, \boldsymbol{U}_I$, let us re-write the random vector $\widetilde{\boldsymbol{z}}$ to illustrate its structure.

[1] This can be shown by writing

$$\begin{aligned}\boldsymbol{U}^H\boldsymbol{U} &= \left(\begin{bmatrix}\boldsymbol{U}_R^T\\ \boldsymbol{U}_R^T\end{bmatrix} - j\begin{bmatrix}\boldsymbol{U}_I^T\\ -\boldsymbol{U}_I^T\end{bmatrix}\right)([\boldsymbol{U}_R \mid \boldsymbol{U}_R] + j\,[\boldsymbol{U}_I \mid -\boldsymbol{U}_I])\\ &= \left(\begin{bmatrix}\boldsymbol{U}_R^T\boldsymbol{U}_R & \boldsymbol{U}_R^T\boldsymbol{U}_R\\ \boldsymbol{U}_R^T\boldsymbol{U}_R & \boldsymbol{U}_R^T\boldsymbol{U}_R\end{bmatrix} + \begin{bmatrix}\boldsymbol{U}_I^T\boldsymbol{U}_I & -\boldsymbol{U}_I^T\boldsymbol{U}_I\\ -\boldsymbol{U}_I^T\boldsymbol{U}_I & \boldsymbol{U}_I^T\boldsymbol{U}_I\end{bmatrix}\right) + j\left(\begin{bmatrix}\boldsymbol{U}_R^T\boldsymbol{U}_I & -\boldsymbol{U}_R^T\boldsymbol{U}_I\\ \boldsymbol{U}_R^T\boldsymbol{U}_I & -\boldsymbol{U}_R^T\boldsymbol{U}_I\end{bmatrix} + \begin{bmatrix}\boldsymbol{U}_I^T\boldsymbol{U}_R & \boldsymbol{U}_I^T\boldsymbol{U}_R\\ -\boldsymbol{U}_I^T\boldsymbol{U}_R & -\boldsymbol{U}_I^T\boldsymbol{U}_R\end{bmatrix}\right).\end{aligned}$$

Then by equating the above to $I + j0$, we arrive at our conclusion.

Consider the matrix $\widehat{\boldsymbol{U}} := [\boldsymbol{U}_R \mid \boldsymbol{U}_I] \in \mathbb{R}^{M\times M}$, which is a scaled unitary matrix (because we can check that $\widehat{\boldsymbol{U}}^T\widehat{\boldsymbol{U}} = \frac{1}{2}\boldsymbol{I}$). Next, consider the random vector $\widehat{\boldsymbol{z}} := \widehat{\boldsymbol{U}}^T\boldsymbol{z}$. Because $\widehat{\boldsymbol{U}}$ is (scaled) unitary and $\boldsymbol{z}$ is composed of i.i.d. zero-mean Gaussian random variables of variance $\frac{1}{M}$, the entries of $\widehat{\boldsymbol{z}}$ are also i.i.d. zero-mean Gaussian random variables, but now with variance $\frac{1}{2M}$. Then, from our definition of $\boldsymbol{U}$ in terms of $\boldsymbol{U}_R$ and $\boldsymbol{U}_I$, for any $l \leq \frac{M}{2}$, we have $\widetilde{z}_l = \widehat{z}_l - j\widehat{z}_{l+\frac{M}{2}}$ and for $l > \frac{M}{2}$, we have $\widetilde{z}_l = \widehat{z}_{l-\frac{M}{2}} + j\widehat{z}_l$. This clearly shows that each of the *first* $\frac{M}{2}$ entries of $\widetilde{\boldsymbol{z}}$ is made up of 2 i.i.d. random variables (one being the real component, the other imaginary), and that the other $\frac{M}{2}$ entries are just complex conjugates of the first $\frac{M}{2}$. Because of this, for $l \leq \frac{M}{2}$, $|\widetilde{z}_l|^2 = |\widetilde{z}_{l+\frac{M}{2}}|^2 = \widehat{z}_l^2 + \widehat{z}_{l+\frac{M}{2}}^2$ is the sum of squares of 2 i.i.d. Gaussian random variables.

From the proof of Theorem 3.1.3, we also denoted

$$Z \;:=\; \widehat{\boldsymbol{\Phi}}^H\widehat{\boldsymbol{\Phi}} - \boldsymbol{I} \;=\; \left(\sum_{l=1}^{M/2} |\widetilde{z}_l|^2 X_l X_l^H - \frac{1}{2}\boldsymbol{I}\right) + \left(\sum_{l=1}^{M/2} |\widetilde{z}_l|^2 \left(X_l X_l^H\right)^* - \frac{1}{2}\boldsymbol{I}\right) \;=:\; Z_1 + Z_2.$$

It is again easy to check that $\mathcal{E}\left[\sum_{l=1}^{M/2}\left(|\widetilde{z}_l|^2 X_l X_l^H\right)\right] = \mathcal{E}\left[\sum_{l=1}^{M/2}\left(|\widetilde{z}_l|^2\left(X_l X_l^H\right)^*\right)\right] = \frac{1}{2}\boldsymbol{I}$. Finally, $\widehat{\boldsymbol{\Phi}}$ has RIP conditioning δ whenever $\|Z\|_S \leq \delta$ with $\|Z\|_S := \sup_{y \text{ is } S\text{-sparse}} \dfrac{y^H Z y}{y^H y}$.

Before moving on to the proof, we first present a lemma regarding the random sequence $|z_l|^2$ that will be useful in the sequel.

Lemma 4. *Suppose for $l = 1, \cdots, \frac{M}{2}$, $|\widetilde{z}_l|^2 = \widehat{z}_l^2 + \widehat{z}_{l+M/2}^2$ where $\widehat{z}_l$ for $l = 1, \cdots, M$ is a sequence of i.i.d. zero-mean Gaussian random variables of variance $\frac{1}{2M}$. Also suppose that $\eta \leq 1$ is a fixed probability. For the random variable $\max_{l=1,\cdots,M/2} |\widetilde{z}_l|^2$, we have the following bounds on the expected value and tail probability of this extreme value:*

$$\mathcal{E}\left[\max_{l=1,\cdots,M/2} |\widetilde{z}_l|^2\right] \;\leq\; \frac{1}{M}\left(\log\left(\frac{C_1 M}{2}\right) + 1\right), \tag{93}$$

$$\mathcal{P}\left[\max_{l=1,\cdots,M/2} |\widetilde{z}_l|^2 > \frac{C_2 \log\left(C_2' M \eta^{-1}\right)}{M}\right] \;\leq\; \eta. \tag{94}$$

Proof. To ease notation, every index l used as a variable for a maximization will be taken over the set $l = 1, \ldots, \frac{M}{2}$ without explicitly writing the index set. To calculate $\mathcal{E}\left[\max_l |\widetilde{z}_l|^2\right]$,

we use the following result that allows us to bound the expected value of a positive random variable by its tail probability (see Proposition 6.1 of [Rauhut]):

$$\mathcal{E}\left[\max_l |\widetilde{z}_l|^2\right] = \int_0^\infty \mathcal{P}\left[\max_l |\widetilde{z}_l|^2 > u\right] du. \tag{95}$$

Using the union bound, we have the estimate $\mathcal{P}\left[\max_l |\widetilde{z}_l|^2 > u\right] \leq \frac{M}{2}\mathcal{P}\left[|\widetilde{z}_1|^2 > u\right]$ (since the $|\widetilde{z}_l|^2$ are identically distributed). Now, because $|\widetilde{z}_1|^2$ is a sum of squares of two Gaussian random variables and thus is a (generalized) χ^2 random variable with 2 degrees of freedom (which we shall denote by χ_2),[2] we have

$$\mathcal{P}\left[|\widetilde{z}_1|^2 > u\right] = \mathcal{P}\left[\chi_2 > 2Mu\right] = \frac{1}{\Gamma(1)} e^{\frac{-2Mu}{2}} = C_1 e^{-Mu},$$

where $\Gamma(\cdot)$ is the Gamma function and the $2Mu$ appears instead of u in the exponential because of the standardization of the Gaussian random variables (initially of variance $\frac{1}{2M}$). To proceed, we break the integral in (95) into 2 parts. To do so, notice that if $u < \frac{1}{M}\log\left(\frac{C_1 M}{2}\right)$, then the trivial upper bound of $\mathcal{P}\left[\max_l |\widetilde{z}_l|^2 > u\right] \leq 1$ is a better estimate than $\frac{C_1 M}{2} e^{-Mu}$. In other words, our estimate for the tail bound of $\max_l |\widetilde{z}_l|^2$ is not very good for small u but gets better with increasing u. Therefore, we have

$$\begin{aligned}
\mathcal{E}\left[\max_l |\widetilde{z}_l|^2\right] &\leq \int_0^{\frac{1}{M}\log\left(\frac{C_1 M}{2}\right)} 1 \, du + \int_{\frac{1}{M}\log\left(\frac{C_1 M}{2}\right)}^\infty \frac{C_1 M}{2} e^{-Mu} \, du \\
&= \frac{1}{M}\log\left(\frac{C_1 M}{2}\right) - \frac{C_1 M}{2}\left[\frac{1}{M} e^{-Mu}\right]\Bigg|_{\frac{1}{M}\log\left(\frac{C_1 M}{2}\right)}^\infty \\
&= \frac{1}{M}\log\left(\frac{C_1 M}{2}\right) + \frac{C_1}{2} e^{-\log\left(\frac{C_1 M}{2}\right)} = \frac{1}{M}\left(\log\left(\frac{C_1 M}{2}\right) + 1\right).
\end{aligned}$$

This is the bound in expectation that we seek for in Equation (95).

In the second part of the proof that follows, C, C' denote universal constants. Essentially, we will want to apply Lemma 3 that is used in Appendix 8.4 to obtain our tail bound. In the lemma, the tail bound of a random variable X can be estimated once we know the

[2] The pdf of a χ^2 random variable χ_q with q degrees of freedom is given by $p(x) = \frac{1}{2^{q/2}\Gamma(q/2)} x^{q/2-1} e^{-x/2}$. Therefore, it's tail probability can be obtained by integration: $\mathcal{P}\left[\chi_q > u\right] = \int_u^\infty p(x)dx$.

moments of X. Therefore, we require the moments of the random variable $\max_l |\widetilde{z}_l|^2$. For this, for any $p > 0$, we use the following simple estimate:

$$\mathcal{E}\left[\max_l |\widetilde{z}_l|^{2p}\right] \leq \frac{M}{2} \max_l \mathcal{E}\left[|\widetilde{z}_l|^{2p}\right] = \frac{M}{2}\mathcal{E}\left[|\widetilde{z}_1|^{2p}\right], \tag{96}$$

where the first step comes from writing the expectation as an integral of the cumulative distribution (as seen in Equation (95)) and taking the union bound, and the second step comes from the fact that the $|\widetilde{z}_l|^2$ are identically distributed. Now, $|\widetilde{z}_1|^2$ is a sub-exponential random variable since it is a sum of squares of Gaussian random variables [239].[3] Therefore, for any $p > 0$, it's p-th moment can be bounded by

$$\mathcal{E}\left[|\widetilde{z}_1|^{2p}\right]^{1/p} \leq \frac{C'}{M} C^{1/p} p,$$

where the division by M comes again from the variance of the Gaussian random variables that make up $|\widetilde{z}_1|^2$. Putting this bound with Equation (96), we have the following estimate for the p-th moments of $\max_l |\widetilde{z}_l|^2$:[4]

$$\mathcal{E}\left[\max_l |\widetilde{z}_l|^{2p}\right]^{1/p} \leq \frac{C'}{M}\left(\frac{CM}{2}\right)^{1/p} p.$$

Therefore, by Lemma 3 with $\alpha = \frac{C'}{M}$, $\beta = \frac{CM}{2}$, and $\gamma = 1$, we have

$$\mathcal{P}\left[\max_l |\widetilde{z}_l|^2 > \frac{eC'u}{M}\right] \leq \frac{CM}{2} e^{-u}.$$

By choosing $u = \log\left(\frac{CM}{2}\eta^{-1}\right)$, we have our desired tail bound of

$$\mathcal{P}\left[\max_l |\widetilde{z}_l|^2 > \frac{C_2 \log\left(C_2' M \eta^{-1}\right)}{M}\right] \leq \eta.$$

□

Armed with this lemma, we can now turn out attention to the main proof. As stated earlier, this follows essentially the same form as [238] with the primary difference of including the results from Lemma 4. As before, because $\mathcal{P}\left[||Z||_S > \delta\right] \leq \mathcal{P}\left[||Z_1||_S > \delta/2\right] +$

[3] A sub-exponential random variable is a random variable whose tail probability is bounded by $\exp^{-Cu}$ for some constant C. Thus, a χ^2 random variable is a specific instance of a sub-exponential random variable.

[4] We remark that this bound gives a worse estimate for the expected value as that calculated before because of the crude bound given by Equation (96).

$\mathcal{P}[||Z_2||_S > \delta/2]$ with $Z_2 = Z_1^*$, we just have to consider bounding the tail bound $\mathcal{P}[||Z_1||_S > \delta/2]$. This proof differs from that in Appendix 8.4 in that here, we will first show that $\mathcal{E}[||Z_1||_S]$ is small when M is large enough and then show that Z_1 does not differ much from $\mathcal{E}[||Z_1||_S]$ with high probability.

Expectation

In this section, we will show that $\mathcal{E}[||Z_1||_S]$ is small. This will basically follow from Lemma 2 in Appendix 8.4 and Equation (93) in Lemma 4. To be precise, the remainder of this section is to prove:

Theorem 7. *Choose any $\delta' \leq \frac{1}{2}$. If $M \geq \frac{C_3 S \mu(\Psi)^2 \log^5 N}{\delta'^2}$, then $\mathcal{E}[||Z||_S] \leq \delta'$.*

Proof. Again, C is some universal constant that may not be the same from line to line. We follow the same symmetrization step found in the proof in Appendix 8.4 to arrive at:

$$E := \mathcal{E}[||Z_1||_S] \leq 2\mathcal{E}\left[\mathcal{E}\left[\left\|\sum_{l=1}^{M/2} \xi_l |\widetilde{z}_l|^2 X_l X_l^H\right\|_S | \{w_l\}, \widetilde{z}\right]\right],$$

where the outer expectation is over the Rademacher sequence $\{\xi_l\}$ and the inner expectation is over the random "frequencies" $\{w_l\}$ and feed-forward vector $\widetilde{z}$. As before, for $l = 1, \cdots, \frac{M}{2}$, we set $Y_l = \widetilde{z}_l^* X_l$. Observe that by definition $\kappa := \max_{l=1,\cdots,M/2} ||Y_l||_\infty = \max_l |\widetilde{z}_l| ||X_l||_\infty$ and thus is a random variable. We then use Lemma 2 with $p = 1$ to get

$$\begin{aligned} E &\leq 2C\sqrt{S}\log(100S)\sqrt{\log(4N)\log(5M)}\mathcal{E}\left[\kappa\sqrt{\left\|\sum_{l=1}^{M/2} |\widetilde{z}_l|^2 X_l X_l^H\right\|_S}\right] \\ &\leq \sqrt{4CS\log^4(N)}\sqrt{\mathcal{E}[\kappa^2]}\sqrt{\mathcal{E}\left[\left\|\sum_{l=1}^{M/2} |\widetilde{z}_l|^2 X_l X_l^H\right\|_S\right]} \\ &\leq \sqrt{4CS\log^4(N)}\sqrt{\mathcal{E}[\kappa^2]}\sqrt{E + \frac{1}{2}}, \end{aligned} \tag{97}$$

where the second line uses the Cauchy-Schwarz inequality for expectations and the third line uses triangle inequality. Again, to get to $\log^4 N$ in the second line, we used our assumption that $N \geq M$, $N \geq S$ and $N \geq O(1)$ in Theorem 3.1.3. It therefore remains to calculate $\mathcal{E}\left[\kappa^2\right]$. Now, $\kappa = \max_l |\widetilde{z}_l| ||X_l||_\infty \leq \max_l |\widetilde{z}_l| \max_l ||X_l||_\infty$. First, we have $\max_l ||X_l||_\infty =$

$\max_{l,n} |\langle \boldsymbol{F}_l, \boldsymbol{\Psi}_n \rangle| \leq \mu(\boldsymbol{\Psi})$. Next, (93) in Lemma 4 tells us that $\mathcal{E}\left[\max_{l=1,\cdots,M/2} |\widetilde{z}_l|^2\right] \leq \frac{1}{M}\left(\log\left(\frac{C_1 M}{2}\right)+1\right)$. Thus, we have $\mathcal{E}\left[\kappa^2\right] \leq \frac{\mu(\boldsymbol{\Psi})^2}{M}\left(\log\left(\frac{C_1 M}{2}\right)+1\right)$. Putting everything together, we have

$$E = \mathcal{E}\left[\|Z_1\|_S\right] \leq \sqrt{\frac{CS\,\log^4(N)\left(\log\left(\frac{C_1 M}{2}\right)+1\right)\mu(\boldsymbol{\Psi})^2}{M}}\sqrt{E+\frac{1}{2}}.$$

Now, the above can be written as $E \leq a\sqrt{E+\frac{1}{2}}$, where $a = \sqrt{\frac{CS\,\log^4(N)\left(\log\left(\frac{C_1 M}{2}\right)+1\right)\mu(\boldsymbol{\Psi})^2}{M}}$. By squaring it, rearranging the terms and completing the squares, we have $E \leq \frac{a^2}{2} + a\sqrt{\frac{1}{2}+\frac{a^2}{4}}$. By supposing $a \leq \frac{1}{2}$, this can be simplified as $E \leq a$. To conclude, let us choose M such that $a \leq \delta'$ where $\delta' \leq \frac{1}{2}$ is our pre-determined conditioning (which incidentally fulfills our previous assumption that $a \leq \frac{1}{2}$). By applying the formula for a, we have that if $M \geq \frac{C_3 S \mu(\boldsymbol{\Psi})^2 \log^5(N)}{\delta'^2}$, then $E \leq \delta'$. □

Tail Probability

To give a probability tail bound estimate to Z_1, we use the following lemma found in [53, 238]:

Lemma 5. *Suppose Y_l for $l = 1, \cdots, M$ are independent, symmetric random variables such that $\|Y_l\|_S \leq \zeta < \infty$ almost surely. Let $Y = \sum_{l=1}^{M} Y_l$. Then for any $u, t > 1$, we have*

$$\mathcal{P}\left[\|Y\|_S > C(u\mathcal{E}\left[\|Y\|_S\right] + t\zeta)\right] \leq e^{-u^2} + e^{-t}.$$

The goal of this section is to prove:

Theorem 8. *Pick any $\delta \leq \frac{1}{2}$ and suppose $N^{-\log^4(N)} \leq \eta \leq \frac{1}{e}$. Suppose $M \geq \frac{C_4 S \mu(\boldsymbol{\Psi})^2 \log^5 N \log \eta^{-1}}{\delta^2}$, then $\mathcal{P}\left[\|Z_1\|_S > \delta\right] \leq 8\eta$.*

Proof. To use Lemma 5, we want Y_l to look like the summands of

$$Z_1 = \sum_{l=1}^{M/2}\left(|\widetilde{z}_l|^2 X_l X_l^H - \mathcal{E}\left[|\widetilde{z}_l|^2 X_l X_l^H\right]\right).$$

However, this poses several problems. First, they are *not* symmetric[5] and thus, we need to symmetrize it by defining

$$\begin{aligned}\widetilde{Y}_l &= |\widetilde{z}_l|^2 X_l X_l^H - |\widetilde{z}'_l|^2 X'_l (X'_l)^H \\ &\sim \xi_l \left(|\widetilde{z}_l|^2 X_l X_l^H - |\widetilde{z}'_l|^2 X'_l (X'_l)^H \right)\end{aligned}$$

where $\widetilde{z}', X'_l$ are independent copies of $\widetilde{z}$ and X_l respectively, and ξ_l is an independent Rademacher sequence. Here, the relation $X \sim Y$ for two random variables X, Y means that X has the same distribution as Y. To form $\widetilde{Y}_l$, what we have done is take each summand of Z_1 and take it's difference with an independent copy of itself. Because $\widetilde{Y}_l$ is symmetric, adding a Rademacher sequence does not change its distribution and this sequence is only introduced to resolve a technicality that will arise later on. If we let $\widetilde{Y} := \sum_{l=1}^{M/2} \widetilde{Y}_l$, then the random variables $\widetilde{Y}$ (symmetrized) and Z_1 (un-symmetrized) are related via the following estimates [53]:

$$\mathcal{E}\left[\|\widetilde{Y}\|_S \right] \leq 2\mathcal{E}\left[\|Z_1\|_S \right], \tag{98}$$

$$\mathcal{P}\left[\|Z_1\|_S > 2\mathcal{E}\left[\|Z_1\|_S \right] + u \right] \leq 2\mathcal{P}\left[\|\widetilde{Y}\|_S > u \right]. \tag{99}$$

However, a second condition imposed on Y_l in Lemma 5 is that $\|Y_l\|_S \leq \zeta < \infty$ almost surely. Because of the unbounded nature of the Gaussian random variables $\widetilde{z}_l$ and $\widetilde{z}'_l$ in $\widetilde{Y}_l$, this condition is not met. Therefore, we need to define a Y_l that is conditioned on the event that these Gaussian random variables are bounded. To do so, define the following event:

$$F = \left\{ \max\left\{ \max_l |\widetilde{z}_l|^2, \ \max_l |\widetilde{z}'_l|^2 \right\} \leq \frac{C_2 \log\left(C'_2 M \eta^{-1} \right)}{M} \right\}.$$

[5] A random variable X is symmetric if X and $-X$ has the same distribution.

Using Equation (94) in Lemma 4, we can calculate $\mathcal{P}[F^c]$, where F^c is the complementary event of F:

$$\begin{aligned}
\mathcal{P}[F^c] &= \mathcal{P}\left[\max\left\{\max_l |\widetilde{z}_l|^2,\ \max_l |\widetilde{z}'_l|^2\right\} > \frac{C_2 \log\left(C'_2 M\eta^{-1}\right)}{M}\right] \\
&\leq \mathcal{P}\left[\max_l |\widetilde{z}_l|^2 > \frac{C_2 \log\left(C'_2 M\eta^{-1}\right)}{M}\right] + \mathcal{P}\left[\max_l |\widetilde{z}'_l|^2 > \frac{C_2 \log\left(C'_2 M\eta^{-1}\right)}{M}\right] \\
&\leq 2\eta.
\end{aligned}$$

Conditioned on event F, the $\|\cdot\|_S$ norm of $\widetilde{Y}_l$ is well-bounded:

$$\begin{aligned}
\left\|\widetilde{Y}_l\right\|_S &= \left\||\widetilde{z}_l|^2 X_l X_l^H - |\widetilde{z}'_l|^2 X'_l (X'_l)^H\right\|_S \ \leq\ 2\max\left\{\max_l |\widetilde{z}_l|^2,\ \max_l |\widetilde{z}'_l|^2\right\}\left\|X_l X_l^H\right\|_S \\
&= \frac{2C_2 \log\left(C'_2 M\eta^{-1}\right)}{M} \sup_{y \text{ is } S\text{-sparse}}\left\{\frac{y^H X_l X_l^H y}{y^H y}\right\} \\
&\leq \frac{2C_2 \log\left(C'_2 M\eta^{-1}\right)}{M} \sup_{y \text{ is } S\text{-sparse}}\left\{\|X_l\|_\infty^2 \frac{\|y\|_1^2}{\|y\|_2^2}\right\} \\
&\leq \frac{2SC_2 \log\left(C'_2 M\eta^{-1}\right)}{M} \max_l \|X_l\|_\infty^2 \ \leq\ \frac{CS\mu(\mathbf{\Psi})^2 \log\left(C'_2 M\eta^{-1}\right)}{M} := \zeta,
\end{aligned}$$

where in the last line we used the fact that the ratio between the ℓ_1 and ℓ_2 norms of an S-sparse vector is S, and the estimate we derived for $\max_l \|X_l\|_\infty^2$ in Appendix 8.4.

We now define a new random variable that is a truncated version of $\widetilde{Y}_l$ which takes for value 0 whenever we fall under event F^c, i.e.,

$$Y_l := \widetilde{Y}_l\, \mathbb{I}_F = \xi_l\left(|\widetilde{z}_l|^2 X_l X_l^H - |\widetilde{z}'_l|^2 X'_l (X'_l)^H\right)\mathbb{I}_F,$$

where $\mathbb{I}_F$ is the indicator function of event F_l. If we define $Y = \sum_{l=1}^{M/2} Y_l$, then the random variables Y (truncated) and $\widetilde{Y}$ (un-truncated) are related by [238] (see also Lemma 1.4.3 of [240])

$$\mathcal{P}\left[\|\widetilde{Y}\|_S > u\right] \leq \mathcal{P}\left[\|Y\|_S > u\right] + \mathcal{P}\left[F^c\right]. \tag{100}$$

When $\widetilde{z}, \widetilde{z}', X_l, X'_l$ are held constant so only the Rademacher sequence ξ_l is random, then the contraction principle [238, 241] tells us that $\mathcal{E}\left[\|Y\|_S\right] \leq \mathcal{E}\left[\|\widetilde{Y}\|_S\right]$. Note that the sole reason

for introducing the Rademacher sequences is for this use of the contraction principle. As this holds point-wise for all $\widetilde{z}, \widetilde{z}', X_l, X_l'$, we have

$$\mathcal{E}\left[\|Y\|_S\right] \leq \mathcal{E}\left[\|\widetilde{Y}\|_S\right]. \tag{101}$$

We now have all the necessary ingredients to apply Lemma 5. First, by choosing $\delta' \leq \frac{1}{2}$, from Theorem 7, we have that $\mathcal{E}\left[\|Z\|_S\right] \leq \delta'$ whenever $M \geq \frac{C_3 S \mu(\mathbf{\Psi})^2 \log^5 N}{\delta'^2}$. Thus, by chaining (101) and (98), we have

$$\mathcal{E}\left[\|Y\|_S\right] \leq \mathcal{E}\left[\|\widetilde{Y}\|_S\right] \leq 2\mathcal{E}\left[\|Z_1\|_S\right] \leq 2\delta'.$$

Also, with this choice of M, we have

$$\zeta = \frac{CS\mu(\mathbf{\Psi})^2 \log\left(C_2' M \eta^{-1}\right)}{M} \leq \frac{C\delta'^2 \log\left(C_2' M \eta^{-1}\right)}{\log^5 N}.$$

Using these estimates for ζ and $\mathcal{E}\left[\|Y\|_S\right]$, and choosing $u = \sqrt{\log \eta^{-1}}$ and $t = \log \eta^{-1}$, Lemma 5 says that

$$\mathcal{P}\left[\|Y\|_S > C'\left(2\delta' \sqrt{\log \eta^{-1}} + \frac{C\delta'^2 \log\left(C_2' M \eta^{-1}\right) \log \eta^{-1}}{\log^5 N}\right)\right] \leq 2\eta.$$

Then, using the relation between the tail probabilities of Y and $\widetilde{Y}$ (100) together with our estimate for $\mathcal{P}\left[F^c\right]$, we have

$$\mathcal{P}\left[\|\widetilde{Y}\|_S > C'\left(2\delta' \sqrt{\log \eta^{-1}} + \frac{C\delta'^2 \log\left(C_2' M \eta^{-1}\right) \log \eta^{-1}}{\log^5 N}\right)\right] \leq 2\eta + \mathcal{P}\left[F^c\right] \leq 4\eta.$$

Finally, using the relation between the tail probabilities of $\widetilde{Y}$ and Z (99), we have

$$\mathcal{P}\left[\|Z_1\|_S > 2\delta' + 2C'\delta' \sqrt{\log \eta^{-1}} + \frac{CC'\delta'^2 \log\left(C_2' M \eta^{-1}\right) \log \eta^{-1}}{\log^5 N}\right] \leq 8\eta,$$

where we used the fact that $\mathcal{E}\left[\|Z_1\|_S\right] \leq \delta'$. Then, for a pre-determined conditioning $\delta \leq \frac{1}{2}$, pick $\delta' = \frac{\delta}{3C'' \sqrt{\log \eta^{-1}}}$ for a constant C'' which will be chosen appropriately later. With this choice of δ' and with our assumptions that $\delta \leq \frac{1}{2}$ and $\eta \leq \frac{1}{e}$, the three terms in the tail

bound becomes

$$\begin{aligned} 2\delta' &= \frac{\delta}{3C''\sqrt{\log \eta^{-1}}} \leq \frac{1}{C''}\frac{\delta}{3}, \\ 2C'\delta'\sqrt{\log \eta^{-1}} &= \frac{2C'}{C''}\frac{\delta}{3}, \\ \frac{CC'\delta'^2 \log\left(C_2' M \eta^{-1}\right)\log \eta^{-1}}{\log^5 N} &= \frac{CC'\delta^2(\log(C_2'M) + \log \eta^{-1})}{9\,(C'')^2 \log^5 N} \\ &\leq \frac{CC'(\log(C_2'M) + \log \eta^{-1})}{3\,(C'')^2 \log^5 N}\frac{\delta}{3}. \end{aligned}$$

As for the last term, if $\eta \geq \frac{1}{C_2' M}$, then $\frac{CC'(\log(C_2'M)+\log \eta^{-1})}{3(C'')^2 \log^5 N} \leq \frac{2CC' \log(C_2'M)}{3(C'')^2 \log^5 N} \leq \frac{2CC'}{3(C'')^2}$ (where we further supposed that $N \geq O(1)$). If $N^{-\log^4 N} \leq \eta \leq \frac{1}{C_2' M}$ (where the lower bound is from the theorem assumptions), then $\frac{CC'(\log(C_2'M)+\log \eta^{-1})}{3(C'')^2 \log^5 N} \leq \frac{2CC' \log \eta^{-1}}{3(C'')^2 \log^5 N} \leq \frac{2CC'}{3(C'')^2}$. By choosing C'' appropriately large, we then have

$$\mathcal{P}\left[\|Z_1\|_S > \frac{\delta}{3} + \frac{\delta}{3} + \frac{\delta}{3}\right] \leq 8\eta.$$

Putting the formula for δ' into $M \geq \frac{C_3 S \mu(\mathbf{\Psi})^2 \log^5 N}{\delta'^2}$ completes the proof. □

8.6 RIP with Multiple Sparse Inputs

In this appendix we show analyze the RIP of networks with multiple input streams, proving Theorem 3. The main approach follows very closely to the proof of Theorem 3.1.3 in Appendix 8.5. As the majority of the proof is identical, we will describe in this appendix the deviations from the previous proof.

Expectation

As in Appendix 8.5, we can define the product $\mathbf{\Phi}^H\mathbf{\Phi}$ as the sum of rank-1 operators, $X_l X_l^H$, where $X_l^H := \sum_{k=1}^L \widetilde{bmZ}_{l,k} \boldsymbol{F}_l^H \mathbf{\Psi}^{k,l}$, the conjugate of the l-th row of $[\widetilde{\boldsymbol{Z}}_1\boldsymbol{F}, \widetilde{\boldsymbol{Z}}_2\boldsymbol{F}, \cdots, \widetilde{\boldsymbol{Z}}_L\boldsymbol{F}]\mathbf{\Psi}$, where $\boldsymbol{F}_l^H := \left[1,\ e^{jw_l},\ \cdots,\ e^{jw_l(N-1)}\right] \in \mathbb{C}^N$ is the conjugated l-th row of $\boldsymbol{F}$. With this small change in the definition of X_l^H, the majority of the proof in Appendix 8.5 holds. In fact all the steps up to Equation (97) can be followed, yielding

$$E \leq \sqrt{C_1 S \log^4(NL)} \sqrt{\mathcal{E}[\kappa^2]} \sqrt{E + \frac{1}{2}} \tag{102}$$

where instead of using $N \geq M$ and $S \geq S$ and $N \geq O(1)$ we use $NL \geq M$ and $NL \geq S$ and $NL \geq O(1)$. From Equation (102) note that the main difference in the expectation bound for the single input model is that the κ, the maximum infinity norm of X_l, bounds a different quantity. Instead of bounding the max-inf of a set of vectors $\max_l |\widetilde{z_l}| \|X_l\|_\infty = \max_l |\widetilde{z_l}| \|\mathbf{\Psi}^H F_l\|_\infty$, we instead need to bound the max inf of sums of vectors $\max_l \sum_{k=1}^{L} \|\widetilde{\mathbf{Z}}_{l,k} \mathbf{\Psi}^{k,l} F_l\|_\infty$.

To replace the bound on $\mathcal{E}(\kappa^2)$ we note that the k^{th} element of X_l is essentially a sum of independent Gaussian random variables, weighted by the inner product $X_l(k) = \sum_{i=1}^{L} \widetilde{z_i} \boldsymbol{F}_l^H \mathbf{\Psi}_k^{l,i}$. κ can then be described as the maximum of $MLN/2$ random variables

$$\kappa = \max_{\substack{1 \leq l \leq M/2 \\ 1 \leq p \leq NL}} |X_l(p)|.$$

To bound this quantity, we can replace the previously used Lemma 4 with the following lemma corollary:

Lemma 6. *Suppose that there are n independent complex Gaussian random variables,* $z_1,\ z_2, \cdots\ z_n$. *And* $z_i = x_i + \mathrm{j}y_i$, *where* x_i *and* y_i *are the real and imaginary part of* z_i. x_i *and* y_i *are independent Gaussian random variables with zero mean and variation* $1/2M$. $\phi_1,\ \phi_2,\ \cdots\ \phi_n$ *are n random variables which are independent of* z_i *and satisfy*

$$\sum_{i=1}^{n} |\phi_i|^2 \leq \mu_0^2.$$

Let $w = \sum_{i=1}^{n} z_i \phi_i$, *then*

$$\mathcal{P}(|w|^2 > u) \leq e^{-\frac{Mu}{\mu_0^2}}. \tag{103}$$

Proof. We use x and y to denote the real and imaginary part of w, a_i and b_i to denote the

real and imaginary part of ϕ_i. There is

$$\begin{aligned} w &= \sum_{i=1}^{n}(x_i + \mathrm{j}y_i)(a_i + \mathrm{j}b_i) \\ &= \sum_{i=1}^{n}(a_i x_i - b_i y_i) + \mathrm{j}\sum_{i=1}^{n}(a_i y_i + b_i x_i) \\ &= x + \mathrm{j}y \end{aligned}$$

When a_i and b_i are considered constant, x and y are Gaussian distributed ($\mathcal{N}(0, \frac{1}{2M}\sum_{i=1}^{n}(a_i^2 + b_i^2))$). Next we need to show that x and y are independent. We note that

$$\begin{aligned} \mathrm{Cov}(a_i x_i - b_i y_i, a_i y_i + b_i x_i | a_i, b_i) &= \mathrm{Cov}(a_i x_i, b_i x_i | a_i, b_i) + \mathrm{Cov}(-b_i y_i, a_i y_i | a_i, b_i) \\ &= \frac{a_i b_i}{2M} - \frac{a_i b_i}{2M} = 0. \end{aligned}$$

With the independence of x_i, y_i and x_j, y_j when $i \neq j$, we know that x and y are independent. Therefore $\frac{2M}{\sum_{i=1}^{n}(a_i^2+b_i^2)}|w|^2$ has a χ^2 distribution when a_i and b_i are regarded as constants. We use χ_2 to denote a χ^2 random variable with 2 degrees of freedom. According to the results of χ^2 distribution, there is

$$\mathcal{P}(|w|^2 > u | a_i, b_i) = \mathrm{P}\left(\chi_2 > \frac{2Mu}{\sum_{i=1}^{n}(a_i^2 + b_i^2)} | a_i, b_i\right) = e^{-\frac{Mu}{\sum_{i=1}^{n}(a_i^2+b_i^2)}} \leq e^{-\frac{Mu}{\mu_0^2}}. \tag{104}$$

Since (104) holds for any possible a_i and b_i, we have

$$\mathcal{P}(|w|^2 > u) \leq e^{-\frac{Mu}{\mu_0^2}}. \tag{105}$$

□

Corollary 1. *For Q independent random variables $w_1, w_2, \cdots, w_Q$, assume $w_i = \sum_{l=1}^{n} z_{i,l}\phi_{il}$, $z_{i,l} = x_{i,l} + \mathrm{j}y_{i,l}$ and all $x_{i,l}$, $y_{i,l}$, $1 \leq i \leq Q$, $1 \leq l \leq n$ are i.i.d Gaussian random variables with zero mean and variance $1/2M$. Suppose that for any i, there is*

$$\sum_{l=1}^{n} |\phi_{i,l}|^2 \leq \mu_0^2.$$

Let $w_{\max} = \max_{1\leq i\leq Q} |w_i|$, then for $\eta > 0$, there is

$$\mathcal{P}(w_{\max}^2 > \frac{\mu_0^2}{M} \ln \frac{Q}{\eta}) \leq \eta,$$

and

$$\mathcal{E}w_{\max}^2 \leq \frac{\mu_0^2}{M}(\ln Q + 1).$$

Proof. According to Lemma 6, by taking a union bound, there is

$$\mathcal{P}(w_{\max}^2 > u) \leq Qe^{-\frac{Mu}{\mu_0^2}}. \tag{106}$$

Let $\eta = Qe^{-\frac{Mu}{\mu_0^2}}$. There is

$$\mathcal{P}(w_{\max}^2 > \frac{\mu_0^2}{M}\ln\frac{Q}{\eta}) \leq \eta. \tag{107}$$

Then we have

$$\begin{aligned}\mathcal{E}w_{max}^2 &= \int_0^\infty \mathrm{P}(w_{max}^2 > u)du \\ &\leq \int_0^{\frac{\mu_0^2}{M}\ln Q} 1du + \int_{\frac{\mu_0^2}{M}\ln Q}^\infty Qe^{-\frac{Mu}{\mu_0^2}}du = \frac{\mu_0^2}{M}(\log Q + 1).\end{aligned}$$

□

By Corollary 1 we can see that we can bound

$$\mathcal{P}(\kappa^2 > \frac{\mu^2(\Psi)}{M}\ln\frac{MNL}{2\eta}) \leq \eta,$$

and

$$\mathcal{E}\kappa^2 \leq \frac{\mu^2(\Psi)}{M}(\ln\frac{MNL}{2} + 1) \leq C_2\frac{\mu^2(\Psi)}{M}\ln(NL),$$

which when combined with Equation (102) gives us that $E < \delta'^2$ when

$$M \geq \frac{C_3 S\mu^2(\mathbf{\Psi})\ln^5(NL)}{\delta'^2}. \tag{108}$$

Tail Probability

With the new expectation bound, the tail bound also closely follows Appendix 8.5. Here we also leverage Lemma 5, however while we have the new, correct expectation bound, we need to also re-derive ζ to match the new matrix structure. We can again symmetrize as

$$Z = \sum_{i=1}^{M/2}\left(X_i X_i^{\mathrm{H}} - X_i' X_i'^{\mathrm{H}}\right). \tag{109}$$

and derive the value ζ that bounds $||Z||_S$ in probability. First we use a Cauchy-Swartz inequality to bound

$$||Z||_S = ||X_i X_i^{\mathrm{H}} - X_i' X_i'^{\mathrm{H}}||_K \leq 2 \max\{||X_i X_i^{\mathrm{H}}||_K, ||X_i' X_i'^{\mathrm{H}}||_K\},$$

and we can bound the maximum of

$$||X_i X_i^{\mathrm{H}}||_K \leq \sup_{y\, is\, S-sparse} ||X_i||_\infty^2 \frac{||y||_1^2}{||y||_2^2} \leq S\kappa^2.$$

From our previous results on bounding κ, we can see that

$$\mathcal{P}\left(||X_i X_i^{\mathrm{H}}||_S > \frac{S\mu^2(\mathbf{\Psi})}{M} \log \frac{MNL}{2\eta}\right) \leq \eta,$$

which gives us the probability that the event $\mathbf{F}$, defined as

$$\mathbf{F} = \left\{\max\{||X_i X_i^{\mathrm{H}}||_S, ||X_i' X_i'^{\mathrm{H}}||_S\} \leq \frac{S\mu^2(\mathbf{\Psi})}{M} \log \frac{MNL}{2\eta} := \zeta\right\}, \tag{110}$$

that our random variables are bounded is $1 - \eta$ (i.e. $\mathrm{P}(\mathbf{F}^C) \leq 2\eta$). Using this new tail probability ζ along with the new expectation bound in Lemma 5 as in Appendix 8.5 yields the desired result, that

$$\mathcal{P}\left(||Z_1||_S > 2\delta' + 2\tilde{C}\delta' \sqrt{\log \eta^{-1}} + \tilde{C} \ln \eta^{-1} \frac{C_4 \delta'^2 \ln\left(\frac{1}{2}NLM\eta^{-1}\right)}{\ln^5(NL)}\right) \leq 8\eta,$$

for constants $\tilde{C}$ and C_4, and Z_1 is as defined in Appendix 8.5. Using the same finishing steps as in Appendix 8.5, we can see that if $(NL)^{-\ln^4(NL)} \leq \eta < 1/e$ then we obtain our desired result, i.e. when

$$M \geq \frac{CS\mu^2(\mathbf{\Psi}) \log^5(NM) \log(\eta^{-1})}{\delta^2}. \tag{111}$$

then

$$\mathcal{P}(||Z_1||_S > \delta) \leq 8\eta,$$

8.7 RIP for Multiple Low-Rank Inputs

In this appendix we prove Theorem 4 that a low-rank input matrix $\boldsymbol{X}$ can be recovered from the network state $\boldsymbol{x}[N]$ via the nuclear norm optimization program (30). To prove this theorem we utilize the concept of the dual certificate, which has been used to prove similar results in [63, 87, 98, 242]. In this methodology we seek a certificate $\boldsymbol{Y}$ whose projections into and out of the space spanned by the singular vectors of $\boldsymbol{X}$ are bounded appropriately. Specifically if we consider the singular value decomposition of $\boldsymbol{X}$ as

$$\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{V}^*$$

and we consider the projection $\mathcal{P}_T$ which projects a matrix into the space T spanned by the left and right singular vectors,

$$\mathcal{P}_T(\boldsymbol{W}) = \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{W} + \boldsymbol{W}\boldsymbol{V}\boldsymbol{V}^* - \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{W}\boldsymbol{V}\boldsymbol{V}^* \tag{112}$$

the conditions for the dual certificate are that $\mathcal{A}$ is injective on T and there exists a matrix $\boldsymbol{Y}$ which satisfies

$$\left\|\mathcal{P}_T(\boldsymbol{Y}) - \boldsymbol{Q}\boldsymbol{V}^H\right\|_F \leq \frac{1}{2\sqrt{2}\gamma} \tag{113}$$

$$\|\mathcal{P}_{T^\perp}(\boldsymbol{Y})\| \leq \frac{1}{2} \tag{114}$$

where the projection $\mathcal{P}_{T^\perp}$ is the projection onto the perpendicular space to T,

$$\mathcal{P}_{T^\perp}(\boldsymbol{W}) = (\boldsymbol{I} - \boldsymbol{Q}\boldsymbol{Q}^*)\,\boldsymbol{W}\,(\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^*)$$

$$\mathcal{A}(\boldsymbol{W}) = \mathrm{vec}\left(\langle \boldsymbol{A}_n, \boldsymbol{W}\rangle\right) \tag{115}$$

The remainder of this proof will be devoted to demonstrating that there does exist a certificate $\boldsymbol{Y}$ by iteratively devising $\boldsymbol{Y}$ via a golfing scheme [98, 242]. The golfing scheme essentially generates an iterative method which defined a series of certificate vectors $\boldsymbol{Y}_k$ for

$k \in [1, \cdots, \kappa]$ which converge to a certificate $\boldsymbol{Y}_\kappa$ which satisfies the necessary conditions. As in [98], we can initialize the 0^{th} iterate to zero, and define the k^{th} iterate in terms of the $\boldsymbol{Y}_{k-1}$ as

$$\boldsymbol{Y}_k = \boldsymbol{Y}_{k-1} + \kappa \mathcal{A}_k^* \mathcal{A}_k (\boldsymbol{Q}\boldsymbol{V}^* - \mathcal{P}_T(\boldsymbol{Y}_{k-1})). \tag{116}$$

We can see that since every iterate has $\mathcal{A}_k^*$ applied to it, every iteration is projected in to the range of $\mathcal{A}^*$, indicating that the final iteration $\boldsymbol{Y}$ will also be in the range of $\mathcal{A}^*$. In [98], Asif and Romberg define a simpler iteration

$$\widetilde{\boldsymbol{Y}}_k = (\mathcal{P}_T - \kappa \mathcal{P}_T \mathcal{A}_k^* \mathcal{A}_k \mathcal{P}_T) \widetilde{\boldsymbol{Y}}_{k-1},$$

which is expressed in terms of the modified certificate

$$\widetilde{\boldsymbol{Y}}_k = \mathcal{P}_T(\boldsymbol{Y}_k) - \boldsymbol{Q}\boldsymbol{V}^*.$$

What remains now is to demonstrate that this iterative procedure converges, with high probability, to a certificate which satisfies the desired dual certificate conditions. We start by using Lemma 7 and observing that the Forbenious norm of the k^{th} iterate is well bounded with probability $1 - O((LN)^{-\beta})$ by

$$\begin{aligned}
\left\| \widetilde{\boldsymbol{Y}}_k \right\|_F &\leq \max_k \left\| \mathcal{P}_T - \kappa \mathcal{P}_T \mathcal{A}_k^* \mathcal{A}_k \mathcal{P}_T \right\| \left\| \widetilde{\boldsymbol{Y}}_{k-1} \right\|_F \\
&\leq 2^{-k} \left\| \widetilde{\boldsymbol{Y}}_0 \right\|_F \\
&\leq 2^{-k} \left\| \boldsymbol{Q}\boldsymbol{V}^* \right\|_F \\
&\leq 2^{-k} \sqrt{R},
\end{aligned}$$

so long that $M \leq c\beta\kappa R(N + \mu_0^2 L) \log^2(LN)$. As in [98] we observe that when we choose $\kappa \geq 0.5 \log_2(8\gamma^2 R)$, the bound for the Frobenious norm of $\widetilde{\boldsymbol{Y}}_\kappa$ is bounded by $\left\| \widetilde{\boldsymbol{Y}}_\kappa \right\|_F \leq (2\sqrt{2}\gamma)^{-1}$.

To show that the second condition on the certificate is also satisfies, we utilize Lemma 8. We begin with writing the quantity we wish to bound in terms of the past golfing scheme

iterate

$$\begin{aligned}
\|\mathcal{P}_{T^\perp}(\boldsymbol{Y}_\kappa)\| &\leq \sum_{k=1}^{\kappa} \left\| \mathcal{P}_{T^\perp}\left(\kappa \mathcal{A}_k^* \mathcal{A}_k \widetilde{\boldsymbol{Y}}_{k-1}\right) \right\| \\
&= \sum_{k=1}^{\kappa} \left\| \mathcal{P}_{T^\perp}\left(\kappa \mathcal{A}_k^* \mathcal{A}_k \widetilde{\boldsymbol{Y}}_{k-1} - \widetilde{\boldsymbol{Y}}_{k-1}\right) \right\| \\
&\leq \sum_{k=1}^{\kappa} \left\| \kappa \mathcal{A}_k^* \mathcal{A}_k \widetilde{\boldsymbol{Y}}_{k-1} - \widetilde{\boldsymbol{Y}}_{k-1} \right\| \\
&\leq \sum_{k=1}^{\kappa} \left\| \kappa \mathcal{A}_k^* \mathcal{A}_k \widetilde{\boldsymbol{Y}}_{k-1} - \widetilde{\boldsymbol{Y}}_{k-1} \right\|_F \\
&\leq \sum_{k=1}^{\kappa} \max_{k\in[1,\ldots\kappa]} \left\| \kappa \mathcal{A}_k^* \mathcal{A}_k \widetilde{\boldsymbol{Y}}_{k-1} - \widetilde{\boldsymbol{Y}}_{k-1} \right\|_F \\
&\leq \sum_{k=1}^{\kappa} \max_{k\in[1,\ldots\kappa]} \frac{1}{2} 2^{-k} \\
&\leq \sum_{k=1}^{\kappa} \frac{1}{2}
\end{aligned} \tag{117}$$

We use Lemma 8 to bound the maximum spectral norm of $\kappa \mathcal{A}_k^* \mathcal{A}_k \widetilde{\boldsymbol{Y}}_{k-1} - \widetilde{\boldsymbol{Y}}_{k-1}$ with probability $1 - O((LN)^{1-\beta})$. Taking $\kappa \geq \log(LN)$ completes the proof.

This bound shows that the final certificate $\boldsymbol{Y}_\kappa$ satisfies all the desired properties. Thus there exists a unique minimum to the nuclear norm optimization program, and the low-rank set of inputs are recoverable from the network node values.

Bound on $\left\|\kappa \mathcal{P}_T \mathcal{A}_k^* \mathcal{A}_k \mathcal{P}_T - \mathcal{P}_T\right\|$

Lemma 7. *Let $\mathcal{P}_T$ be defined as in Equation* (112) *and $\mathcal{A}_k$ be the restricted measurement operator as defined in Equation* (115)*. Then if the number of nodes scale as*

$$M \geq c\beta\kappa R\left(N + \mu_0^2 L\right)\log^2(LN)$$

for a constant $\beta > 1$, then with probability greater then $1 - O(\kappa(LN)^{-\beta}$, we have

$$\max_{k\in[1,\ldots,\kappa]} \|\kappa \mathcal{P}_T \mathcal{A}^* \mathcal{A} \mathcal{P}_T - \mathcal{P}_T\| \leq \frac{1}{2}$$

Proof. This lemma bounds the operator norm

$$\left\|\kappa\mathcal{P}_T\mathcal{A}_k^*\mathcal{A}_k\mathcal{P}_T - \mathcal{P}_T\right\|.$$

Since $\mathcal{E}\left[\mathcal{A}_k^*\mathcal{A}_k\right] = \frac{1}{\kappa}\mathcal{I}$, this norm is equivalent to

$$\begin{aligned}\kappa\mathcal{P}_T\mathcal{A}_k^*\mathcal{A}_k\mathcal{P}_T - \mathcal{P}_T &= \kappa\mathcal{P}_T\mathcal{A}_k^*\mathcal{A}_k\mathcal{P}_T - \mathcal{E}\left[\kappa\mathcal{P}_T\mathcal{A}_k^*\mathcal{A}_k\mathcal{P}_T\right] \\ &= \kappa\sum_{n\in\Gamma_k}\left(\mathcal{P}_T(\boldsymbol{A}_n)\otimes\mathcal{P}_T(\boldsymbol{A}_n) - \mathcal{E}\left[\mathcal{P}_T(\boldsymbol{A}_n)\otimes\mathcal{P}_T(\boldsymbol{A}_n)\right]\right)\end{aligned}$$

We can also define here $\mathcal{L}_n(\boldsymbol{C}) = \langle\mathcal{P}_T(\boldsymbol{A}_n), \boldsymbol{C}\rangle\mathcal{P}_T(\boldsymbol{A}_n)$ which has $\|\mathcal{L}_n\| = \|\mathcal{P}_T(\boldsymbol{A}_n)\|_F^2$ which gives us

$$\kappa\mathcal{P}_T\mathcal{A}_k^*\mathcal{A}_k\mathcal{P}_T - \mathcal{E}\left[\kappa\mathcal{P}_T\mathcal{A}_k^*\mathcal{A}_k\mathcal{P}_T\right] = \kappa\sum_{n\in\Gamma_k}(\mathcal{L}_n - \mathcal{E}\left[\mathcal{L}_n\right])$$

To calculate the variance, we can use the symmetry of $\mathcal{L}_n$ to only calculate

$$\begin{aligned}\kappa^2\left\|\sum_{n\in\Gamma_k}\mathcal{E}\left[\mathcal{L}_n^2\right] - \mathcal{E}\left[\mathcal{L}_n\right]^2\right\| &\leq \kappa^2\left\|\sum_{n\in\Gamma_k}\mathcal{E}\left[\mathcal{L}_n^2\right]\right\| \\ &= \kappa^2\left\|\mathcal{E}\left[\sum_{n\in\Gamma_k}\|\mathcal{P}_T(\boldsymbol{A}_n)\|_F^2\mathcal{L}_n\right]\right\|\end{aligned}$$

We now need to bound $\|\mathcal{P}_T(\boldsymbol{A}_n)\|_F^2$, which can be done by the following:

$$\begin{aligned}\|\mathcal{P}_T(\boldsymbol{A}_n)\|_F^2 &= \langle\mathcal{P}_T(\boldsymbol{A}_n), \boldsymbol{A}_n\rangle \\ &= \langle\boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{z}_n\boldsymbol{f}_n^*, \boldsymbol{z}_n\boldsymbol{f}_n^*\rangle + \langle\boldsymbol{z}_n\boldsymbol{f}_n^*\boldsymbol{V}\boldsymbol{V}^*, \boldsymbol{z}_n\boldsymbol{f}_n^*\rangle - \langle\boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{z}_n\boldsymbol{f}_n^*\boldsymbol{V}\boldsymbol{V}^*, \boldsymbol{z}_n\boldsymbol{f}_n^*\rangle \\ &= \|\boldsymbol{f}_n\|_2^2\|\boldsymbol{Q}^*\boldsymbol{z}_n\|_2^2 + \|\boldsymbol{z}_n\|_2^2\|\boldsymbol{V}^*\boldsymbol{f}_n\|_2^2 - \|\boldsymbol{Q}^*\boldsymbol{z}_n\|_2^2\|\boldsymbol{V}^*\boldsymbol{f}_n\|_2^2 \\ &\leq N\|\boldsymbol{Q}^*\boldsymbol{z}_n\|_2^2 + \|\boldsymbol{z}_n\|_2^2\|\boldsymbol{V}^*\boldsymbol{f}_n\|_2^2\end{aligned}$$

Using this we can write

$$\begin{aligned}\left\|\mathcal{E}\left[\sum_{n\in\Gamma_k}\|\mathcal{P}_T\left(\boldsymbol{A}_n\right)\|_F^2\,\mathcal{L}_n\right]\right\| &\leq \left\|\sum_{n\in\Gamma_k}\mathcal{E}\left[\left(N\|\boldsymbol{Q}^*\boldsymbol{z}_n\|_2^2 + \|\boldsymbol{z}_n\|_2^2\|\boldsymbol{V}^*\boldsymbol{f}_n\|_2^2\right)\mathcal{L}_n\right]\right\| \\ &\leq N\left\|\mathcal{E}\left[\sum_{n\in\Gamma_k}\|\boldsymbol{Q}^*\boldsymbol{z}_n\|_2^2\mathcal{L}_n\right]\right\| + \left\|\mathcal{E}\left[\sum_{n\in\Gamma_k}\|\boldsymbol{z}_n\|_2^2\|\boldsymbol{V}^*\boldsymbol{f}_n\|_2^2\mathcal{L}_n\right]\right\| \\ &\leq N\left\|\sum_{n\in\Gamma_k}\mathcal{E}\left[\|\boldsymbol{Q}^*\boldsymbol{z}_n\|_2^2\mathcal{L}_n\right]\right\| + \sup\|\boldsymbol{V}^*\boldsymbol{f}_n\|_\infty\left\|\sum_{n\in\Gamma_k}\mathcal{E}\left[\|\boldsymbol{z}_n\|_2^2\mathcal{L}_n\right]\right\| \\ &\leq N\left\|\sum_{n\in\Gamma_k}\mathcal{E}\left[\|\boldsymbol{Q}^*\boldsymbol{z}_n\|_2^2\mathcal{L}_n\right]\right\| + R\mu_0^2\left\|\sum_{n\in\Gamma_k}\mathcal{E}\left[\|\boldsymbol{z}_n\|_2^2\mathcal{L}_n\right]\right\|\end{aligned}$$

We now need to bound these two quantities. First we look to bound the first quantity

$$\begin{aligned}\left\|\sum_{n\in\Gamma_k}\mathcal{E}\left[\|\boldsymbol{Q}^*\boldsymbol{z}_n\|_2^2(\mathcal{P}_T(\boldsymbol{A}_n)\otimes\mathcal{P}_T(\boldsymbol{A}_n))\right]\right\| &\leq \|\mathcal{P}_T\|\left\|\mathcal{E}\left[\|\boldsymbol{Q}^*\boldsymbol{z}_n\|_2^2(\boldsymbol{A}_n\otimes\boldsymbol{A}_n)\right]\right\|\|\mathcal{P}_T\| \\ &\leq \left\|\mathcal{E}\left[\|\boldsymbol{Q}^*\boldsymbol{z}_n\|_2^2(\boldsymbol{A}_n\otimes\boldsymbol{A}_n)\right]\right\| \\ &\leq \left\|\mathcal{E}\left[\|\boldsymbol{Q}^*\boldsymbol{z}_n\|_2^2\{z_n[\alpha]z_n^*[\beta]\boldsymbol{f}_n\boldsymbol{f}_n^*\}_{\alpha,\beta}\right]\right\|\end{aligned}$$

Expanding, we have:

$$\begin{aligned}* &= \left\|\mathcal{E}\left[\left(\sum_{l=1}^{L}\|\boldsymbol{q}_l\|_2^2|z_n[l]|^2+2\sum_{l\neq m}\mathrm{Re}(\langle\boldsymbol{q}_l,\boldsymbol{q}_m\rangle z_n[l]z_n^*[m])\right)z_n[\alpha]z_n^*[\beta]\boldsymbol{I}_N\right]_{\alpha,\beta}\right\| \\ &= \left\|\left\{\left(\frac{1}{M^2}\sum_{\alpha=1}^{L}\|\boldsymbol{q}_\alpha\|_2^2+2\sum_{l\neq m}\mathcal{E}\left[\mathrm{Re}(\langle\boldsymbol{q}_l,\boldsymbol{q}_m\rangle z_n[l]z_n^*[m])z_n[\alpha]z_n^*[\beta]\right]\right)\boldsymbol{I}_N\right\}_{\alpha,\beta}\right\| \\ &= \left\|\left\{\frac{1}{M^2}\sum_{l=1}^{L}\|\boldsymbol{q}_l\|_2^2\boldsymbol{I}_N\delta_{\alpha=\beta}+\frac{2}{M^2}\langle\boldsymbol{q}_\alpha,\boldsymbol{q}_\beta\rangle\boldsymbol{I}_N\delta_{\alpha\neq\beta}\right\}_{\alpha,\beta}\right\| \\ &= \frac{1}{M^2}\left\|\left\{\|\boldsymbol{Q}\|_F^2\boldsymbol{I}_N\delta_{\alpha=\beta}+2\langle\boldsymbol{q}_\alpha,\boldsymbol{q}_\beta\rangle\boldsymbol{I}_N\delta_{\alpha\neq\beta}\right\}_{\alpha,\beta}\right\|\end{aligned}$$

giving us

$$\begin{aligned}\left\|\sum_{n\in\Gamma_k}\mathcal{E}\left[\|\boldsymbol{Q}^*\boldsymbol{z}_n\|_2^2(\mathcal{P}_T(\boldsymbol{A}_n)\otimes\mathcal{P}_T(\boldsymbol{A}_n))\right]\right\| &\leq \frac{1}{M\kappa}\|\boldsymbol{Q}\|_F^2+\|\boldsymbol{Q}\boldsymbol{Q}^*\| \\ &\leq \frac{R+1}{M\kappa}\end{aligned}$$

Similarly, for the second term we can take $\boldsymbol{Q}=\boldsymbol{I}_L$ to get

$$\begin{aligned}\left\|\sum_{n\in\Gamma_k}\mathcal{E}\left[\|\boldsymbol{z}_n\|_2^2(\mathcal{P}_T(\boldsymbol{A}_n)\otimes\mathcal{P}_T(\boldsymbol{A}_n))\right]\right\| &\leq \frac{1}{M\kappa}\|\boldsymbol{I}\|_F^2 \\ &= \frac{L}{M\kappa}\end{aligned}$$

Putting the pieces together, we get

$$\sigma_X^2=\kappa R\frac{N+\mu_0^2L}{M}$$

To use the matrix Bernstein inequality, it now remains to bound the Orlicz-1 norm $\kappa\|\mathcal{L}_n-\mathcal{E}[\mathcal{L}_n]\|_{\psi_1}$. We can use the PSD quality of $\mathcal{L}_n$ and its expectation to obtain

$$\|\mathcal{L}_n-\mathcal{E}[\mathcal{L}_n]\|_{\psi_1}\leq\max\left\{\|\mathcal{L}_n\|_{\psi_1}-\|\mathcal{E}[\mathcal{L}_n]\|_{\psi_1}\right\}$$

The norm of $\|\mathcal{E}[\mathcal{L}_n]\|$ can be calculated via

$$\|\mathcal{E}[\mathcal{L}_n]\| = \|\mathcal{E}[\mathcal{P}_T(\boldsymbol{A}_n)]\|_F^2 = \mathcal{E}\left[\sum_m |z_n[m]|^2 |f_n[m]|^2\right] = \frac{1}{M}\|\boldsymbol{f}_n\|_2^2 = \frac{N}{M},$$

which indicates that the second of these two terms is simply $\|\mathcal{E}[\mathcal{L}_n]\|_{\psi_1} = N/(M\log(2))$. To calculate $\|\mathcal{E}[\mathcal{L}_n]\|_{\psi_1}$, we use the definition of the Orlitcz-1 norm:

$$\|\mathcal{E}[\mathcal{L}_n]\|_{\psi_1} = \inf\left\{y : \mathcal{E}\left[e^{\|\mathcal{L}_n\|/y}\right] < 2\right\}$$

$$\begin{aligned}
\|\mathcal{E}[\mathcal{L}_n]\|_{\psi_1} &= \left\|\|\mathcal{P}_T(\boldsymbol{A}_n)\|_2^2\right\|_{\psi_1} \\
&\leq \left\|N\|\boldsymbol{Q}^* z_n\|_2^2 + \|z_n\|_2^2\|\boldsymbol{V}^* \boldsymbol{f}_n\|_2^2\right\|_{\psi_1} \\
&\leq \left\|N\|\boldsymbol{Q}^* z_n\|_2^2 + RN\mu_0^2\|z_n\|_2^2\right\|_{\psi_1} \\
&\leq N\left\|\|\boldsymbol{Q}^* z_n\|_2^2\right\|_{\psi_1} + RN\mu_0^2\left\|\|z_n\|_2^2\right\|_{\psi_1}
\end{aligned}$$

Using the result in Equation (121) with $\sigma^2 = 1/M$ in the first term and in $\sigma^2 = R/M$ in the second term yields

$$\begin{aligned}
\|\mathcal{E}[\mathcal{L}_n]\|_{\psi_1} &\leq N\left\|\|\boldsymbol{Q}^* z_n\|_2^2\right\|_{\psi_1} + RN\mu_0^2\left\|\|z_n\|_2^2\right\|_{\psi_1} \\
&\leq 2\frac{N}{2M(1-4^{-\frac{1}{R}})} + 2\frac{R\mu_0^2}{2M(1-4^{-\frac{1}{L}})} \\
&\leq \frac{1}{M}\left(\frac{N}{M(1-4^{-\frac{1}{R}})} + \frac{R\mu_0^2}{1-4^{-\frac{1}{L}}}\right) \\
&\leq \frac{2}{\log(2)M}\left(NR + LR\mu_0^2\right) \\
&\leq \frac{2R\left(N + L\mu_0^2\right)}{\log(2)M}
\end{aligned}$$

We now have appropriate bounds on both the variance and the Orliscz norm, which

allows us to bound the largest singular value using the Matrix Bernstein inequality Specifically, we can see that the first term in Theorem 9 is bounded as

$$\sigma_X \sqrt{t + \log(L+N)} \leq \sqrt{\kappa R \frac{N + \mu_0^2 L}{M} (t + \log(L+N))} \tag{118}$$

Letting $t = \beta \log(LN) > \log(N+L)$ gives

$$\sigma_X \sqrt{t + \log(L+N)} \leq \sqrt{2\kappa R\beta \frac{N + \mu_0^2 L}{M} \log(LN)}$$

Likewise we can bound the second term:

$$\begin{aligned}
U_1 \log\left(\frac{MU_1^2}{\sigma_X^2}\right)(t + \log(L+N)) &\leq 2\beta U_1 \log\left(\frac{MU_1^2}{\sigma_X^2}\right)\log(LN) \\
&\leq 2\beta U_1 \log\left(\frac{4\Delta\kappa R(N+\mu_0^2 L)}{\log^2(2)M}\right)\log(LN) \\
&\leq \frac{8\beta\kappa R(N+\mu_0^2 L)}{\log(2)M} \log\left(R(N+\mu_0^2 L)\right)\log(LN) \\
&\leq c\frac{\beta\kappa R(N+\mu_0^2 L)}{M} \log\left(R(N+\mu_0^2 L)\right)\log(LN) \\
&\leq c\frac{\beta\kappa R(N+\mu_0^2 L)}{M} \log^2(LN)
\end{aligned}$$

Thus to appropriately bound

$$\left\|\kappa \mathcal{P}_T \mathcal{A}_k^* \mathcal{A}_k \mathcal{P}_T - \mathcal{P}_T\right\| \leq c \max\left\{\sqrt{\frac{\kappa R\beta(N+\mu_0^2 L)}{M}\log(LN)}, \frac{\beta\kappa R(N+\mu_0^2 L)}{M}\log^2(LN)\right\}$$

we can see that we would need

$$M \geq C\beta\kappa R(N+\mu_0^2 L)\log^2(LN)$$

and taking the union bound over the κ partitions completes the proof of the lemma.

□

Bound on $\|(\mathcal{A}^*\mathcal{A} - \mathcal{I})(\boldsymbol{G})\|$

Lemma 8. *Let $\mathcal{A}_k$ be defined as in Equation* (115)*, $\kappa < M$ be the number of steps in the golfing scheme and assume that $M \leq LN$. Then as long as*

$$M \geq c\beta\kappa \max\left(N + \mu_0^2 L\right) \log^2(NL),$$

where μ_k^2 is the coherence term defined by

$$\mu_k^2 = R \sup_{\omega \in [0,2\pi]} \left\| \widetilde{\boldsymbol{Y}}_k^* \boldsymbol{f}_\omega \right\|_2^2, \tag{119}$$

, then with probability at least $1 - O(M(LN)^{-\beta})$, we have

$$\max_k \left\| \kappa \mathcal{A}_k^* \mathcal{A}_k(\widetilde{\boldsymbol{Y}}_{k-1}) - \widetilde{\boldsymbol{Y}}_{k-1} \right\| \leq 2^{-(k+1)}.$$

Proof. Lemma 9 essentially bounds the operator norm of $\kappa\mathcal{A}^*\mathcal{A} - \mathcal{I}$. In particular, to prove Theorem 2, the reduced version with $\kappa = 1$ is needed. Lemma 8 uses the matrix Bernstein inequality to accomplish this task, taking

$$X_n = \kappa(\langle \boldsymbol{G}, \boldsymbol{A}_n \rangle \boldsymbol{A}_n - \mathcal{E}\left[\langle \boldsymbol{G}, \boldsymbol{A}_n \rangle \boldsymbol{A}_n\right])$$

and we just need to control $\left\| \sum \mathcal{E}\left[X_n X_n^*\right] \right\|$ and $\left\| \sum \mathcal{E}\left[X_n^* X_n\right] \right\|$. To bound the second of these, we can calculate

$$\begin{aligned}
\left\| \sum_{n \in \Gamma_k} \mathcal{E}\left[X_n^* X_n\right] \right\| &\leq \kappa^2 \left\| \sum_{n \in \Gamma_k} \mathcal{E}\left[|\langle \boldsymbol{G}, \boldsymbol{A}_n \rangle|^2 \boldsymbol{A}_n \boldsymbol{A}_n^* \right] \right\| \\
&= \kappa^2 \left\| \sum_{n \in \Gamma_k} \mathcal{E}\left[|\langle \boldsymbol{G}, \boldsymbol{A}_n \rangle|^2 \boldsymbol{z}_n \boldsymbol{f}_n^* \boldsymbol{f}_n \boldsymbol{z}_n^* \right] \right\| \\
&= \kappa^2 \left\| \sum_{n \in \Gamma_k} \mathcal{E}\left[\left\| \boldsymbol{f}_n \right\|_2^2 |\langle \boldsymbol{G}, \boldsymbol{A}_n \rangle|^2 \boldsymbol{z}_n \boldsymbol{z}_n^* \right] \right\| \\
&= N\kappa^2 \left\| \sum_{n \in \Gamma_k} \mathcal{E}\left[|\langle \boldsymbol{G}, \boldsymbol{A}_n \rangle|^2 \boldsymbol{z}_n \boldsymbol{z}_n^* \right] \right\| \\
&\leq \frac{3N\kappa^2}{M^2} \|\boldsymbol{G}\|_F^2 \left\| \sum_{n \in \Gamma_k} \boldsymbol{I}_M \right\| \\
&= \frac{3N\kappa}{M} \|\boldsymbol{G}\|_F^2
\end{aligned}$$

where the third inequality is due to Lemma 9. For the other expectation

$$\begin{aligned}
\left\|\sum_{n\in\Gamma_k}\mathcal{E}\left[X_nX_n^*\right]\right\| &\leq \kappa^2\left\|\sum_{n\in\Gamma_k}\mathcal{E}\left[|\langle \boldsymbol{G},\boldsymbol{A}_n\rangle|^2\boldsymbol{A}_n^*\boldsymbol{A}_n\right]\right\| \\
&= \kappa^2\left\|\sum_{n\in\Gamma_k}\mathcal{E}\left[|\langle \boldsymbol{G},\boldsymbol{A}_n\rangle|^2\boldsymbol{f}_n\boldsymbol{z}_n^*\boldsymbol{z}_n\boldsymbol{f}_n^*\right]\right\| \\
&= \kappa^2\left\|\sum_{n\in\Gamma_k}\mathcal{E}\left[\|\boldsymbol{z}_n\|_2^2\,|\langle \boldsymbol{G},\boldsymbol{A}_n\rangle|^2\boldsymbol{f}_n\boldsymbol{f}_n^*\right]\right\| \\
&= \frac{L\kappa^2}{M^2}\left\|\sum_{n\in\Gamma_k}\mathcal{E}\left[\|\boldsymbol{G}\boldsymbol{f}_n\|_2^2\boldsymbol{f}_n\boldsymbol{f}_n^*\right]\right\| \\
&\leq \frac{L\kappa^2}{M^2}\mu\left\|\sum_{n\in\Gamma_k}\sup_{\omega}(\|\boldsymbol{G}\boldsymbol{f}_n\|_\infty)\mathcal{E}\left[|\boldsymbol{f}_n\boldsymbol{f}_n^*|\right]\right\| \\
&= \frac{L\kappa^2}{M^2}\sup_{\omega}(\|\boldsymbol{Q}\Lambda\boldsymbol{V}^*\boldsymbol{f}_n\|_\infty^2)\left\|\sum_{n\in\Gamma_k}\mathbf{1}_N\right\| \\
&\leq \frac{L\kappa}{M}\|\boldsymbol{V}\|_F^2\|\Lambda\|_F^2\sup_{\omega}(\|\boldsymbol{V}^*\boldsymbol{f}_n\|_\infty^2) \\
&\leq \frac{L\kappa}{MR}\|\boldsymbol{Q}\|_F^2\|\boldsymbol{G}\|_F^2\mu^2 \\
&= \frac{L\kappa}{M}\mu^2\|\boldsymbol{G}\|_F^2
\end{aligned}$$

Using these bounds, with $\kappa = 1$ we can write

$$\sigma_X^2 \leq \frac{\kappa}{M}\|\boldsymbol{G}\|_F^2\max\{\mu_0 L, 3N\}$$

and to use Proposition 1 we just need to bound $\|X\|_{\psi_2}$. To start, we can see that

$$
\begin{aligned}
U_1 &= \|X\|_{\psi_1} \leq 2\kappa \left\| |\langle \boldsymbol{G}, \boldsymbol{A}_n \rangle| \boldsymbol{A}_n \right\|_{\psi_1} \\
&\leq 2\kappa \left\| |\langle \boldsymbol{G}, \boldsymbol{A}_n \rangle| \, \|\boldsymbol{A}_n\|_F \right\|_{\psi_1} \\
&\leq c\kappa \, \|\langle \boldsymbol{G}, \boldsymbol{A}_n \rangle\|_{\psi_2} \, \left\| \|\boldsymbol{A}_n\|_F \right\|_{\psi_2} \\
&\leq c\kappa \, \|\langle \boldsymbol{G}, \boldsymbol{A}_n \rangle\|_{\psi_2} \, \sqrt{\left\| \|\boldsymbol{A}_n\|_F^2 \right\|_{\psi_1}} \\
&\leq c\kappa \, \|\langle \boldsymbol{G}, \boldsymbol{A}_n \rangle\|_{\psi_2} \, \sqrt{\left\| \left\| \boldsymbol{f}_n \right\|_2^2 \|\boldsymbol{z}_n\|_2^2 \right\|_{\psi_2}} \\
&= c\kappa N \, \|\langle \boldsymbol{G}, \boldsymbol{A}_n \rangle\|_{\psi_2} \, \sqrt{\left\| \|\boldsymbol{z}_n\|_2^2 \right\|_{\psi_2}} \\
&= c\kappa \sqrt{\frac{N}{M\left(1 - 4^{-1/L}\right)}} \, \|\langle \boldsymbol{G}, \boldsymbol{A}_n \rangle\|_{\psi_2} \\
&= c\kappa \sqrt{\frac{N}{M\left(1 - 4^{-1/L}\right)}} \left\| |\mathrm{trace}(\boldsymbol{f}_n \boldsymbol{z}_n^* \boldsymbol{G})| \right\|_{\psi_2} \\
&= c\kappa \sqrt{\frac{N}{M\left(1 - 4^{-1/L}\right)}} \left\| |\mathrm{trace}(\boldsymbol{z}_n^* \boldsymbol{Q} \boldsymbol{\Lambda} \boldsymbol{V}^* \boldsymbol{f}_n)| \right\|_{\psi_2} \\
&= c\kappa \sqrt{\frac{N}{M\left(1 - 4^{-1/L}\right)}} \left\| \|\boldsymbol{\Lambda}\|_F \left\| \boldsymbol{V}^* \boldsymbol{f}_n \right\|_2 \|\boldsymbol{Q}^* \boldsymbol{z}_n\|_2 \right\|_{\psi_2} \\
&\leq c\kappa \sqrt{\frac{N}{M\left(1 - 4^{-1/L}\right)}} \|\boldsymbol{\Lambda}\|_F \left\| \|\boldsymbol{V}^* \boldsymbol{f}_n\|_2 \|\boldsymbol{Q}^* \boldsymbol{z}_n\|_2 \right\|_{\psi_2} \\
&\leq c\kappa \sqrt{\frac{N\mu_0^2}{MR\left(1 - 4^{-1/L}\right)}} \|\boldsymbol{G}\|_F \, \left\| \|\boldsymbol{Q}^* \boldsymbol{z}_n\|_2 \right\|_{\psi_2} \\
&\leq c\kappa \sqrt{\frac{N\mu_0^2}{MR\left(1 - 4^{-1/L}\right)}} \|\boldsymbol{G}\|_F \sqrt{\left\| \|\boldsymbol{Q}^* \boldsymbol{z}_n\|_2^2 \right\|_{\psi_2}} \\
&\leq c\kappa \sqrt{\frac{N\mu_0^2}{MR\left(1 - 4^{-1/L}\right)}} \|\boldsymbol{G}\|_F \sqrt{\frac{1}{M\left(1 - 4^{-1/R}\right)}} \\
&\leq c\kappa \sqrt{\frac{N\mu_0^2 \|\boldsymbol{G}\|_F^2}{M^2 R\left(1 - 4^{-1/L}\right)\left(1 - 4^{-1/R}\right)}} \\
&\leq c\kappa \sqrt{\frac{NL\mu_0^2 \|\boldsymbol{G}\|_F^2}{M^2}}
\end{aligned}
$$

We can now apply the matrix Bernstein theorem with the calculated values of U_1 and

σ_X. Again using $t = \beta \log(LN)$, the first portion of the bound is

$$\begin{aligned}
\sigma_X \sqrt{t + \log(L+N)} &\leq \|\boldsymbol{G}\|_F \sqrt{\frac{\kappa}{M} \max\{\mu_k^2 L, N\}(\beta \log(LN) + \log(L+N))} \\
&\leq c \|\boldsymbol{G}\|_F \sqrt{\frac{\kappa\beta}{M} \max\{\mu_k^2 L, N\} \log(LN)}
\end{aligned}$$

and the second portion of the bound is

$$\begin{aligned}
U_1 \log\left(\frac{\Delta U_1^2}{\sigma_X}\right)(t + \log(L+N)) &\leq U_1 \log\left(\frac{\Delta U_1^2}{\sigma_X}\right)(\beta \log(LN) + \log(L+N)) \\
&\leq c \|\boldsymbol{G}\|_F \kappa \sqrt{\frac{LN\mu_k^2}{M}} \log\left(c \|\boldsymbol{G}\|_F^2 \kappa^2 \frac{LN\mu_k^2}{M^2} \frac{M}{\kappa \|G\|_F^2 \max\{\mu_k^2 L, N\}}\right) \beta \log(LN) \\
&\leq c \|\boldsymbol{G}\|_F \kappa \sqrt{\frac{LN\mu_k^2}{M}} \log\left(c\Delta\kappa \frac{LN\mu_k^2}{M} \frac{1}{\max\{\mu_k^2 L, N\}}\right) \beta \log(LN) \\
&\leq c\beta \|\boldsymbol{G}\|_F \kappa \sqrt{\frac{LN\mu_k^2}{M}} \log\left(c \frac{LN\mu_k^2}{\max\{\mu_k^2 L, N\}}\right) \log(LN) \\
&\leq c\beta \|\boldsymbol{G}\|_F \kappa \sqrt{\frac{LN\mu_k^2}{M}} \log\left(\min\{\mu_k^2 L, N\}\right) \log(LN)
\end{aligned}$$

This yields a bound of

$$\begin{aligned}
\|(\mathcal{A}^*\mathcal{A} - \mathcal{I})\boldsymbol{G}\| \leq c \|\boldsymbol{G}\|_F \max\Bigg\{ &\sqrt{\frac{\kappa\beta \log(LN)}{M} \log(\max\{\mu_k^2 L, N\})}, \\
&\sqrt{\mu_k^2 LN} \frac{\beta\kappa}{M} \log(LN) \log(\min\{\mu_k^2 L, N\})\Bigg\}
\end{aligned}$$

We can now use Lemma 10 to bound $\mu_k^2 \leq \mu_0^2$ with probability $1 - O(M(LN)^{-\beta})$ and Lemma 7 to bound $\|\boldsymbol{G}_k\|_F \leq 2^{-k}\sqrt{R}$, which gives us

$$\begin{aligned}
\|(\mathcal{A}^*\mathcal{A} - \mathcal{I})\boldsymbol{G}\| \leq c2^{-k/2} \max\Bigg\{ &\sqrt{\frac{\kappa\beta R \log(LN)}{M} \log(\max\{\mu_0^2 L, N\})}, \\
&\sqrt{\mu_0^2 LN} \frac{\beta\kappa}{M} \log(LN) \log(\min\{\mu_0^2 L, N\})\Bigg\}
\end{aligned}$$

Or, simplifying the bound using $R \leq \min\{L, N\}$,

$$\begin{aligned}
\|(\mathcal{A}^*\mathcal{A} - \mathcal{I})\boldsymbol{G}\| \leq c2^{-k/2} \max\Bigg\{ &\sqrt{\frac{\kappa\beta R \log(LN)}{M} \log(LN)}, \\
&\sqrt{\mu_0^2 LN} \frac{\beta\kappa}{M} \log^2(LN))\Bigg\}
\end{aligned}$$

Taking

$$M \geq c\beta\kappa R \max\{N, L\mu_0^2\} \log^2(LN),$$

proves the lemma. To simplify the bound on the probability, we note that Lemma 10 holds with probability $1 - O(M(LN)^{-\beta})$ and this lemma holds with probability $1 - O(\kappa(LN)^{-\beta})$. Since $\kappa < M$ and assuming that $M \leq LN$, we can write that the result holds with probability $1 - O((LN)^{1-\beta})$. Additionally, since Lemma 10 holds when

$$\begin{aligned} M &\geq c\beta\kappa R \left(N + L\mu_0^2\right) \log^2(LN), \\ &\geq c\beta\kappa R \max\{N, L\mu_0^2\} \log^2(LN), \end{aligned}$$

Then both lemmas hold under the same condition.

□

Bound on $\mathcal{E}\left[|\langle \boldsymbol{C}, \boldsymbol{A}_n\rangle|^2 \boldsymbol{z}_n \boldsymbol{z}_n^*\right]$

Lemma 9 essentially bounds the spectrum of the expected matrix

$$\mathcal{E}\left[|\langle \boldsymbol{G}, \boldsymbol{A}_n\rangle|^2 \boldsymbol{z}_n \boldsymbol{z}_n^*\right]:$$

Lemma 9. *Suppose $\boldsymbol{A}_n = \boldsymbol{z}_n \boldsymbol{f}_n^*$ be defined as the outer product of an* i.i.d. *random Gaussian vector $\boldsymbol{z}_n$ with zero mean and variance $1/M$ and a random Fourier vector $\boldsymbol{f}_n$. Then the operator $|\langle \boldsymbol{C}, \boldsymbol{A}_n\rangle|^2 \boldsymbol{z}_n \boldsymbol{z}_n^*$ satisfies*

$$\mathcal{E}_z\left[|\langle \boldsymbol{C}, \boldsymbol{A}_n\rangle|^2 \boldsymbol{z}_n \boldsymbol{z}_n^*\right] \preceq \frac{3}{M^2} \|\boldsymbol{C}^* \boldsymbol{f}_n\|_2^2 \boldsymbol{I}_M$$

and

$$\mathcal{E}_{z,f}\left[|\langle \boldsymbol{C}, \boldsymbol{A}_n\rangle|^2 \boldsymbol{z}_n \boldsymbol{z}_n^*\right] \preceq \frac{3}{M^2} \|\boldsymbol{C}\|_F^2 \boldsymbol{I}_M$$

Proof. To begin the proof, we look at the expectation of each element of the matrix. We

first calculate the expected value with respect to $\boldsymbol{z}_n$,

$$\begin{aligned}
\cdot &= \mathcal{E}_z\left[\left|\sum_{l=1}^{L} z_n[l]\boldsymbol{c}_l^*\boldsymbol{f}_n\right|^2 z_n[\alpha]z_n^*[\beta]\right] \\
&= \mathcal{E}_z\left[\left(\sum_{l=1}^{L} z_n[l]\boldsymbol{c}_l^*\boldsymbol{f}_n\right)^*\left(\sum_{l=1}^{L} z_n[l]\boldsymbol{c}_l^*\boldsymbol{f}_n\right) z_n[\alpha]z_n^*[\beta]\right] \\
&= \mathcal{E}_z\left[\sum_{l=1}^{L}|z_n[l]|^2|\boldsymbol{c}_l^*\boldsymbol{f}_n|^2 z_n[\alpha]z_n^*[\beta] + 2\sum_{k\neq l}\mathrm{Re}\left(z_n^*[l]z_n[k]\langle\boldsymbol{c}_l^*\boldsymbol{f}_n, \boldsymbol{c}_k^*\boldsymbol{f}_n\rangle\right) z_n[\alpha]z_n^*[\beta]\right] \\
&= \left(\frac{3}{2M^2}|\boldsymbol{c}_\alpha^*\boldsymbol{f}_n|^2 + \frac{1}{M^2}\|\boldsymbol{C}\boldsymbol{f}_n\|_2^2\right)\delta_{\alpha=\beta} + \frac{2}{M^2}\langle\boldsymbol{c}_\alpha^*\boldsymbol{f}_n, \boldsymbol{c}_\beta^*\boldsymbol{f}_n\rangle\delta_{\alpha\neq\beta}
\end{aligned}$$

We can then use the matrix formulation

$$\begin{aligned}
\mathcal{E}_z\left[|\langle\boldsymbol{C}, \boldsymbol{A}_n\rangle|^2\boldsymbol{z}_n\boldsymbol{z}_n^*\right] &= \frac{3}{2M^2}\mathrm{diag}(\boldsymbol{C}\boldsymbol{f}_n\boldsymbol{f}_n^*\boldsymbol{C}^*) + \frac{1}{M^2}\|\boldsymbol{C}\boldsymbol{f}_n\|_2^2\boldsymbol{I}_M + \frac{2}{M^2}\boldsymbol{C}\boldsymbol{f}_n\boldsymbol{f}_n^*\boldsymbol{C}^* + \frac{2}{M^2}\mathrm{diag}(\boldsymbol{C}\boldsymbol{f}_n\boldsymbol{f}_n^*\boldsymbol{C}^*) \\
&= \frac{1}{M^2}\|\boldsymbol{C}\boldsymbol{f}_n\|_2^2\boldsymbol{I}_M + 2\boldsymbol{C}\boldsymbol{f}_n\boldsymbol{f}_n^*\boldsymbol{C}^* - \frac{1}{2}\mathrm{diag}(\boldsymbol{C}\boldsymbol{f}_n\boldsymbol{f}_n^*\boldsymbol{C}^*) \\
&\leq \frac{1}{M^2}\|\boldsymbol{C}\boldsymbol{f}_n\|_2^2\boldsymbol{I}_M + 2\boldsymbol{C}\boldsymbol{f}_n\boldsymbol{f}_n^*\boldsymbol{C}^* \\
&\leq \frac{3}{M^2}\|\boldsymbol{C}\boldsymbol{f}_n\|_2^2\boldsymbol{I}_M
\end{aligned}$$

where to obtain the result we first use the linearity of the expectation with the fact that $\mathrm{diag}(\boldsymbol{C}\boldsymbol{f}_n\boldsymbol{f}_n^*\boldsymbol{C}^*)$ is positive-semidefinite, proving the fist portion of the Lemma. To prove the second portion we simply take an expectation with respect to $\boldsymbol{f}_n$:

$$\begin{aligned}
\mathcal{E}_{z,f}\left[|\langle\boldsymbol{C}, \boldsymbol{A}_n\rangle|^2\boldsymbol{z}_n\boldsymbol{z}_n^*\right] &\leq \mathcal{E}_f\left[\frac{3}{M^2}\|\boldsymbol{C}\boldsymbol{f}_n\|_2^2\boldsymbol{I}_M\right] \\
&\leq \frac{3}{M^2}\|\boldsymbol{C}\|_F^2\boldsymbol{I}_M
\end{aligned}$$

thus completing the proof. This gives us the desired property

□

Contractive property of μ_k^2

Lemma 10. *Let μ_k^2 be the coherence factor as defined in Equation* (119)*, and additionally assume that $L > 1$ and that $LN > R\mu_0^4$. If*

$$M \geq c\beta\kappa R\left(N + L\mu_0^2\right)\log^2(LN),$$

then with probability at least $1 - O(\kappa(LN)^{-\beta})$*,*

$$\mu_k^2 \leq 2^{-1}\mu_{k-1}^2,$$

for all $k \in [1, \cdots, \kappa]$

Proof. In this lemma we would like to show that the coherence term at each golfing iteration

$$\begin{aligned}
\mu_k^2 &= \frac{1}{R}\sup_{\omega}\sum_{l=1}^{L}\langle\widetilde{\boldsymbol{Y}}_k, \boldsymbol{e}_l\boldsymbol{f}^*\rangle^2 \\
&= \frac{1}{R}\sup_{\omega}\sum_{l=1}^{L}\langle(\kappa\mathcal{P}_T\mathcal{A}_k^*\mathcal{A}_k\mathcal{P}_T - \mathcal{P}_T)\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{e}_l\boldsymbol{f}^*\rangle^2 \\
&= \frac{1}{R}\sup_{\omega}\sum_{l=1}^{L}\left(\sum_{n\in\Gamma_k}\kappa\langle\mathcal{P}_T(\boldsymbol{A}_n), \boldsymbol{e}_l\boldsymbol{f}^*\rangle\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle - \langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{e}_l\boldsymbol{f}^*\rangle\right)^2 \\
&= \frac{1}{R}\sup_{\omega}\sum_{l=1}^{L}\left(\sum_{n\in\Gamma_k}\kappa\langle\mathcal{P}_T(\boldsymbol{A}_n), \boldsymbol{e}_l\boldsymbol{f}^*\rangle\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle - \mathcal{E}\left[\kappa\langle\mathcal{P}_T(\boldsymbol{A}_n), \boldsymbol{e}_l\boldsymbol{f}^*\rangle\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle\right]\right)^2
\end{aligned} \tag{120}$$

In order to bound this quantity we use the scalar Bernstein inequality on each of the inner quantities

$$\sum_{n\in\Gamma_k} X_n = \sum_{n\in\Gamma_k}\kappa\langle\mathcal{P}_T(\boldsymbol{A}_n), \boldsymbol{e}_l\boldsymbol{f}^*\rangle\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle - \mathcal{E}\left[\kappa\langle\mathcal{P}_T(\boldsymbol{A}_n), \boldsymbol{e}_l\boldsymbol{f}^*\rangle\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle\right]$$

As in the matrix Bernstein formulation, we need to find both the variance and Orlicz norm. First we find the variance,

$$\begin{aligned}
\sum_{n\in\Gamma_k}\mathcal{E}[X_nX_n^*] &= \kappa^2\sum_{n\in\Gamma_k}\mathcal{E}\left[|\langle\mathcal{P}_T(\boldsymbol{A}_n), \boldsymbol{e}_l\boldsymbol{f}^*\rangle|^2|\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle|^2\right] - |\mathcal{E}\left[\langle\mathcal{P}_T(\boldsymbol{A}_n), \boldsymbol{e}_l\boldsymbol{f}^*\rangle\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle\right]|^2 \\
&\leq \kappa^2\sum_{n\in\Gamma_k}\mathcal{E}\left[|\langle\mathcal{P}_T(\boldsymbol{A}_n), \boldsymbol{e}_l\boldsymbol{f}^*\rangle|^2|\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle|^2\right] \\
&= \kappa^2\sum_{n\in\Gamma_k}\mathcal{E}\left[|\langle\boldsymbol{Q}\boldsymbol{Q}^*z_n\boldsymbol{f}_n^*, \boldsymbol{e}_l\boldsymbol{f}^*\rangle + \langle z_n\boldsymbol{f}_n^*\boldsymbol{V}\boldsymbol{V}, \boldsymbol{e}_l\boldsymbol{f}^*\rangle + \langle\boldsymbol{Q}\boldsymbol{Q}^*z_n\boldsymbol{f}_n^*\boldsymbol{V}\boldsymbol{V}^*, \boldsymbol{e}_l\boldsymbol{f}^*\rangle|^2|\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle|^2\right] \\
&\leq \kappa^2\sum_{n\in\Gamma_k}\mathcal{E}\left[\left(|\langle\boldsymbol{Q}\boldsymbol{Q}^*z_n\boldsymbol{f}_n^*, \boldsymbol{e}_l\boldsymbol{f}^*\rangle|^2 + |\langle z_n\boldsymbol{f}_n^*\boldsymbol{V}\boldsymbol{V}^*, \boldsymbol{e}_l\boldsymbol{f}^*\rangle|^2 + |\langle\boldsymbol{Q}\boldsymbol{Q}^*z_n\boldsymbol{f}_n^*\boldsymbol{V}\boldsymbol{V}^*, \boldsymbol{e}_l\boldsymbol{f}^*\rangle|^2\right)|\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle|^2\right] \\
&\leq \kappa^2\sum_{n\in\Gamma_k}\mathcal{E}\left[\left(|\langle\boldsymbol{Q}\boldsymbol{Q}^*z_n\boldsymbol{f}_n^*, \boldsymbol{e}_l\boldsymbol{f}^*\rangle|^2 + |\langle\boldsymbol{f}_n^*\boldsymbol{V}\boldsymbol{V}^*, \boldsymbol{f}^*\rangle z_n[l]|^2 + |\langle\boldsymbol{Q}\boldsymbol{Q}^*z_n\boldsymbol{f}_n^*\boldsymbol{V}\boldsymbol{V}^*, \boldsymbol{e}_l\boldsymbol{f}^*\rangle|^2\right)|\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle|^2\right.
\end{aligned}$$

This sum consists of three terms, the first of which can be bounded by again leveraging Lemma 9,

$$\begin{aligned}
\sum_{n\in\Gamma_k}\mathcal{E}\left[|\langle \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{z}_n\boldsymbol{f}_n^*, \boldsymbol{e}_l\boldsymbol{f}^*\rangle|^2|\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle|^2\right] &= \sum_{n\in\Gamma_k}\mathcal{E}\left[|\boldsymbol{f}_n^*\boldsymbol{f}\langle\boldsymbol{q}_l, \boldsymbol{Q}^*\boldsymbol{z}_n\rangle|^2|\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle|^2\right]\\
&\leq \sum_{n\in\Gamma_k}\mathcal{E}\left[\boldsymbol{f}_n^*\boldsymbol{f}\boldsymbol{f}^*\boldsymbol{f}_n\boldsymbol{q}_l^*\boldsymbol{Q}^*\boldsymbol{z}_n\boldsymbol{z}_n^*\boldsymbol{Q}\boldsymbol{q}_l|\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle|^2\right]\\
&\leq \sum_{n\in\Gamma_k}\mathcal{E}\left[\boldsymbol{f}^*\boldsymbol{f}_n\boldsymbol{f}_n^*\boldsymbol{f}\boldsymbol{q}_l^*\boldsymbol{Q}^*|\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle|^2\boldsymbol{z}_n\boldsymbol{z}_n^*\boldsymbol{Q}\boldsymbol{q}_l\right]\\
&\leq 3\sum_{n\in\Gamma_k}\mathcal{E}\left[\boldsymbol{f}^*\boldsymbol{f}_n\boldsymbol{f}_n^*\boldsymbol{f}\boldsymbol{q}_l^*\boldsymbol{Q}^*\left\|\widetilde{\boldsymbol{Y}}_{k-1}\boldsymbol{f}_n\right\|_2^2\boldsymbol{I}_L\boldsymbol{Q}\boldsymbol{q}_l\right]\\
&= \frac{3}{M^2}\|\boldsymbol{q}_l\|_2^2\sum_{n\in\Gamma_k}\mathcal{E}\left[\boldsymbol{f}^*\boldsymbol{f}_n\boldsymbol{f}_n^*\boldsymbol{f}\left\|\widetilde{\boldsymbol{Y}}_{k-1}\boldsymbol{f}_n\right\|_2^2\right]\\
&\leq \frac{3R\mu_{k-1}^2}{M^3}\|\boldsymbol{q}_l\|_2^2\sum_{n\in\Gamma_k}\boldsymbol{f}^*\mathcal{E}\left[\boldsymbol{f}_n\boldsymbol{f}_n^*\right]\boldsymbol{f}\\
&\leq \frac{3R\mu_{k-1}^2}{M^3}\|\boldsymbol{q}_l\|_2^2\sum_{n\in\Gamma_k}\boldsymbol{f}^*\boldsymbol{I}_N\boldsymbol{f}\\
&= \frac{3R\mu_{k-1}^2}{M^3}\|\boldsymbol{q}_l\|_2^2|\Gamma_k|N\\
&= \frac{3NR\mu_{k-1}^2}{\kappa M^2}\|\boldsymbol{q}_l\|_2^2
\end{aligned}$$

For the second term we have

$$\begin{aligned}
\sum_{n\in\Gamma_k}\mathcal{E}\left[|\langle \boldsymbol{f}_n^*\boldsymbol{V}\boldsymbol{V}^*, \boldsymbol{f}^*\rangle|^2|z_n[l]|^2|\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle|^2\right] &= \sum_{n\in\Gamma_k}\mathcal{E}\left[|\langle\boldsymbol{V}^*\boldsymbol{f}_n, \boldsymbol{V}^*\boldsymbol{f}\rangle|^2|z_n[l]|^2|\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle|^2\right]\\
&= \sum_{n\in\Gamma_k}\mathcal{E}\left[|\langle\boldsymbol{V}^*\boldsymbol{f}_n, \boldsymbol{V}^*\boldsymbol{f}\rangle|^2|\langle\widetilde{\boldsymbol{Y}}_{k-1}, z_n[l]\boldsymbol{A}_n\rangle|^2\right]\\
&= \sum_{n\in\Gamma_k}\mathcal{E}\left[\boldsymbol{f}^*\boldsymbol{V}\boldsymbol{V}^*\boldsymbol{f}_n\boldsymbol{f}_n^*\boldsymbol{V}\boldsymbol{V}^*\boldsymbol{f}|\langle\widetilde{\boldsymbol{Y}}_{k-1}, z_n[l]\boldsymbol{z}_n\boldsymbol{f}_n^*\rangle|^2\right]
\end{aligned}$$

Using the fact that $|z_n[l]|^2 = \boldsymbol{e}_l^*\boldsymbol{z}_n\boldsymbol{z}_n^*\boldsymbol{e}_l$ and Lemma 9, we obtain

$$
\begin{aligned}
\sum_{n\in\Gamma_k}\mathcal{E}\left[|\langle \boldsymbol{f}_n^*\boldsymbol{V}\boldsymbol{V}^*,\boldsymbol{f}^*\rangle|^2|z_n[l]|^2|\langle\widetilde{\boldsymbol{Y}}_{k-1},\boldsymbol{A}_n\rangle|^2\right] &\leq \frac{3}{M^2}\sum_{n\in\Gamma_k}\mathcal{E}\left[\boldsymbol{f}^*\boldsymbol{V}\boldsymbol{V}^*\boldsymbol{f}_n\boldsymbol{f}_n^*\boldsymbol{V}\boldsymbol{V}^*\boldsymbol{f}\boldsymbol{e}_l^*\left\|\widetilde{\boldsymbol{Y}}_{k-1}\boldsymbol{f}_n\right\|_2^2\boldsymbol{I}_L\boldsymbol{e}_l\right]\\
&\leq \frac{3}{M^2}\sum_{n\in\Gamma_k}\mathcal{E}\left[\boldsymbol{f}^*\boldsymbol{V}\boldsymbol{V}^*\boldsymbol{f}_n\boldsymbol{f}_n^*\boldsymbol{V}\boldsymbol{V}^*\boldsymbol{f}\left\|\widetilde{\boldsymbol{Y}}_{k-1}\boldsymbol{f}_n\right\|_2^2\right]\\
&\leq \frac{3R\mu_{k-1}^2}{M^2}\sum_{n\in\Gamma_k}\mathcal{E}\left[\boldsymbol{f}^*\boldsymbol{V}\boldsymbol{V}^*\boldsymbol{f}_n\boldsymbol{f}_n^*\boldsymbol{V}\boldsymbol{V}^*\boldsymbol{f}\right]\\
&\leq \frac{3R\mu_{k-1}^2}{M^2}\sum_{n\in\Gamma_k}\boldsymbol{f}^*\boldsymbol{V}\boldsymbol{V}^*\boldsymbol{f}\\
&\leq \frac{3R\mu_{k-1}^2}{M^2}|\Gamma_k|\,\|\boldsymbol{V}^*\boldsymbol{f}\|_2^2\\
&\leq \frac{3R\mu_{k-1}^2}{\kappa M^1}\,\|\boldsymbol{V}^*\boldsymbol{f}\|_2^2\\
&\leq \frac{3R^2\mu_{k-1}^2\mu_0^2}{\kappa M^1}
\end{aligned}
$$

Finally, for the third term, we have

$$
\begin{aligned}
\sum_{n\in\Gamma_k}\mathcal{E}\left[|\langle \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{z}_n\boldsymbol{f}_n^*\boldsymbol{V}\boldsymbol{V}^*,\boldsymbol{e}_l\boldsymbol{f}^*\rangle|^2|\langle\widetilde{\boldsymbol{Y}}_{k-1},\boldsymbol{A}_n\rangle|^2\right] &= \sum_{n\in\Gamma_k}\mathcal{E}\left[|\langle \boldsymbol{V}^*\boldsymbol{f}_n,\boldsymbol{V}^*\boldsymbol{f}\rangle|^2|\langle \boldsymbol{q}_l,\boldsymbol{Q}^*\boldsymbol{z}_n\rangle|^2|\langle\widetilde{\boldsymbol{Y}}_{k-1},\boldsymbol{A}_n\rangle|^2\right]\\
&= \sum_{n\in\Gamma_k}\mathcal{E}\left[\boldsymbol{f}^*\boldsymbol{V}\boldsymbol{V}^*\boldsymbol{f}_n\boldsymbol{f}_n^*\boldsymbol{V}\boldsymbol{V}^*\boldsymbol{f}|\langle \boldsymbol{q}_l,\boldsymbol{Q}^*\boldsymbol{z}_n\rangle|^2|\langle\widetilde{\boldsymbol{Y}}_{k-1},\boldsymbol{A}_n\rangle|^2\right]\\
&= \sum_{n\in\Gamma_k}\mathcal{E}\left[\|\boldsymbol{V}^*\boldsymbol{f}\|_2^2\,|\langle \boldsymbol{q}_l,\boldsymbol{Q}^*\boldsymbol{z}_n\rangle|^2|\langle\widetilde{\boldsymbol{Y}}_{k-1},\boldsymbol{A}_n\rangle|^2\right]\\
&\leq \sum_{n\in\Gamma_k}\mathcal{E}\left[\|\boldsymbol{V}^*\boldsymbol{f}\|_2^2\,\boldsymbol{q}_l^*\boldsymbol{Q}^*\boldsymbol{z}_n\boldsymbol{z}_n^*|\langle\widetilde{\boldsymbol{Y}}_{k-1},\boldsymbol{A}_n\rangle|^2\boldsymbol{Q}\boldsymbol{q}_l\right]\\
&\leq \frac{3}{M^2}\sum_{n\in\Gamma_k}\mathcal{E}\left[\|\boldsymbol{V}^*\boldsymbol{f}\|_2^2\,\boldsymbol{q}_l^*\boldsymbol{Q}^*\boldsymbol{I}_L\left\|\widetilde{\boldsymbol{Y}}_{k-1}\boldsymbol{f}_n\right\|_2^2\boldsymbol{Q}\boldsymbol{q}_l\right]\\
&\leq \frac{3}{M^2}\sum_{n\in\Gamma_k}\mathcal{E}\left[\|\boldsymbol{V}^*\boldsymbol{f}\|_2^2\,\boldsymbol{q}_l^*\boldsymbol{q}_l\left\|\widetilde{\boldsymbol{Y}}_{k-1}\boldsymbol{f}_n\right\|_2^2\right]\\
&\leq \frac{3}{M^2}\,\|\boldsymbol{q}_l\|_2^2\,\|\boldsymbol{V}^*\boldsymbol{f}\|_2^2\sum_{n\in\Gamma_k}\mathcal{E}\left[\left\|\widetilde{\boldsymbol{Y}}_{k-1}\boldsymbol{f}_n\right\|_2^2\right]\\
&\leq \frac{3R^2}{M^2}\,\|\boldsymbol{q}_l\|_2^2\,\mu_0^2\mu_{k-1}^2|\Gamma_k|\\
&\leq \frac{3R^2\mu_0^2\mu_{k-1}^2}{\kappa M}\,\|\boldsymbol{q}_l\|_2^2
\end{aligned}
$$

Summing these three together, and using $\|\boldsymbol{q}_l\| \leq 1$ yields

$$\sigma_X^2 \leq 9\kappa\left(\frac{NR\mu_{k-1}^2}{M^2}\left\|\boldsymbol{q}_l\right\|_2^2 + 2\frac{R^2\mu_0^2\mu_{k-1}^2}{M}\right)$$

All that remains to use the Bernstein inequality is to find the Orlicz-1 norm of X_n. First, from Lemma 8 we have that

$$\begin{aligned}
\left\|\langle\widetilde{\boldsymbol{Y}}_{k-1}, \boldsymbol{A}_n\rangle\right\|_{\psi_2}^2 &= \left\|\boldsymbol{z}_n^*\widetilde{\boldsymbol{Y}}_{k-1}\boldsymbol{f}_n\right\|_{\psi_2}^2 \\
&\leq \left\|\left\|\boldsymbol{Q}^*\boldsymbol{z}_n\right\|_2\left\|\boldsymbol{\Lambda}_{\boldsymbol{k-1}}\boldsymbol{V}^*\boldsymbol{f}_n\right\|_2\right\|_{\psi_2}^2 \\
&\leq c\frac{R\mu_{k-1}^2}{M}
\end{aligned}$$

For the first term we have

$$\begin{aligned}
\left\|\boldsymbol{f}_n^*\boldsymbol{f}\langle\boldsymbol{q}_l, \boldsymbol{Q}^*\boldsymbol{z}_n\rangle\right\|_{\psi_2}^2 &\leq \left\|\left\|\boldsymbol{f}_n^*\right\|_2\|\boldsymbol{f}\|_2\,|\langle\boldsymbol{q}_l, \boldsymbol{Q}^*\boldsymbol{z}_n\rangle|\right\|_{\psi_2}^2 \\
&\leq \left\|N\left\|\boldsymbol{q}_l\right\|_2\left\|\boldsymbol{Q}^*\boldsymbol{z}_n\right\|_2\right\|_{\psi_2}^2 \\
&\leq N^2\left\|\boldsymbol{q}_l^*\boldsymbol{Q}^*\boldsymbol{z}_n\right\|_{\psi_2}^2 \\
&\leq cN^2\left(\left\|\boldsymbol{q}_l\right\|_2\frac{1}{M}\right)^2 \\
&\leq c\frac{N^2\left\|\boldsymbol{q}_l\right\|_2^2}{M^2}
\end{aligned}$$

For the second term we have

$$\begin{aligned}
\left\|\langle\boldsymbol{V}^*\boldsymbol{f}_n, \boldsymbol{V}^*\boldsymbol{f}\rangle z_n[l]\right\|_{\psi_2}^2 &\leq \left\|\left\|\boldsymbol{V}^*\boldsymbol{f}_n\right\|_2\|\boldsymbol{V}^*\boldsymbol{f}\|_2\, z_n[l]\right\|_{\psi_2}^2 \\
&\leq R^2\mu_0^4\left\|z_n[l]\right\|_{\psi_2}^2 \\
&\leq c\frac{R^2}{M}\mu_0^4
\end{aligned}$$

And for the final term we have

$$\begin{aligned}
\left\|\langle\boldsymbol{V}^*\boldsymbol{f}_n, \boldsymbol{V}^*\boldsymbol{f}\rangle\langle\boldsymbol{q}_l, \boldsymbol{Q}^*\boldsymbol{z}_n\rangle\right\|_{\psi_2}^2 &\leq \left\|\left\|\boldsymbol{V}^*\boldsymbol{f}_n\right\|_2\|\boldsymbol{V}^*\boldsymbol{f}\|_2\,\boldsymbol{q}_l^*\boldsymbol{Q}^*\boldsymbol{z}_n\right\|_{\psi_2}^2 \\
&\leq R^2\mu_0^4\left\|\boldsymbol{q}_l^*\boldsymbol{Q}^*\boldsymbol{z}_n\right\|_{\psi_2}^2 \\
&\leq c\frac{R^2}{M}\mu_0^4\left\|\boldsymbol{q}_l\right\|_2^2
\end{aligned}$$

Now we can calculate the total Orlicz norm via

$$\begin{aligned}
\|X_n\|_{\psi_1}^2 &\leq c\kappa^2 \frac{R\mu_{k-1}^2}{M}\left(\frac{N^2\|\boldsymbol{q}_l\|_2^2}{M^2} + \frac{R^2}{M}\mu_0^4 + \frac{R^2}{M}\mu_0^4\|\boldsymbol{q}_l\|_2^2\right) \\
&\leq c\kappa^2 \frac{R\mu_{k-1}}{M^2}\left(\frac{N^2\|\boldsymbol{q}_l\|_2^2}{M^2} + \frac{2R^2}{M}\mu_0^4\right) \\
&\leq c\frac{\kappa^2 RN^2}{M^3}\|\boldsymbol{q}_l\|_2^2\mu_{k-1}^2 + c\frac{2\kappa^2R^3}{M^2}\mu_0^4\mu_{k-1}^2
\end{aligned}$$

Since we wish to bound the square of the sum of terms, we calculate the square values of the two terms in the Bernstein inequality. The first term is bounded by

$$\begin{aligned}
t\sigma_X^2 &\leq ct\kappa\frac{R}{M}\mu_{k-1}^2\left(\frac{N}{M}\|\boldsymbol{q}_l\|_2^2 + 2R\mu_0^2\right) \\
&\leq c\beta\kappa\frac{R}{M}\mu_{k-1}^2\left(\frac{N}{M}\|\boldsymbol{q}_l\|_2^2 + 2R\mu_0^2\right)\log(LN)
\end{aligned}$$

and the second term is bounded by

$$\begin{aligned}
t^2U_\alpha^2\log^2\left(\frac{|\Gamma_k|U_\alpha^2}{\sigma_X^2}\right) &\leq t^2U_\alpha^2\log^2\left(c\frac{|\Gamma_k|M\kappa^2R\mu_{k-1}^2\left(\frac{N^2}{M}\|\boldsymbol{q}_l\|_2^2 + 2R^2\mu_0^4\right)}{M^2\kappa R\mu_{k-1}^2\left(\frac{N}{M}\|\boldsymbol{q}_l\|_2^2 + 2R\mu_0^2\right)}\right) \\
&\leq t^2c\kappa^2\frac{R}{M^2}\mu_{k-1}^2\left(\frac{N^2}{M}\|\boldsymbol{q}_l\|_2^2 + 2R^2\mu_0^4\right)\log^2\left(c\frac{\frac{N^2}{M}\|\boldsymbol{q}_l\|_2^2 + 2R^2\mu_0^4}{\frac{N}{M}\|\boldsymbol{q}_l\|_2^2 + 2R\mu_0^2}\right) \\
&\leq t^2c\kappa^2\frac{R}{M^2}\mu_{k-1}^2\left(\frac{N^2}{M}\|\boldsymbol{q}_l\|_2^2 + 2R^2\mu_0^4\right)\log^2\left(c\frac{N^2\|\boldsymbol{q}_l\|_2^2 + 2MR^2\mu_0^4}{N\|\boldsymbol{q}_l\|_2^2 + 2RM\mu_0^2}\right)
\end{aligned}$$

Assuming that $L > 1$ and $LN > R\mu_0^4$ gives

$$\begin{aligned}
* &\leq t^2c\kappa^2\frac{R}{M^2}\mu_{k-1}^2\left(\frac{N^2}{M}\|\boldsymbol{q}_l\|_2^2 + 2R^2\mu_0^4\right)\log^2(LN) \\
&\leq c\beta^2\kappa^2\frac{R}{M^2}\mu_{k-1}^2\left(\frac{N^2}{M}\|\boldsymbol{q}_l\|_2^2 + 2R^2\mu_0^4\right)\log^4(LN)
\end{aligned}$$

Each summand is then bounded by the maximum of these two quantities with probability $1 - O(|\Gamma_k|(LN)^{-\beta})$, the $|\Gamma_k|$ term coming from the union bound over all terms in each inner sum.

Using this bound on each summand, we obtain the total bound by taking a union bound, summing over $l \in [1, \cdots, L]$, and dividing by R, yielding a bound of the maximum of

$$t\sigma_X^2 \leq c\beta\kappa\frac{R}{M}\mu_{k-1}^2\left(\frac{N}{M} + 2L\mu_0^4\right)\log(LN)$$

and

$$t^2 U_\alpha^2 \log^2\left(\frac{|\Gamma_k| U_\alpha^2}{\sigma_X^2}\right) \leq \beta^2 c\kappa^2 \frac{R}{M^2}\mu_{k-1}^2 \left(\frac{N}{M} + 2RL\mu_0^2\right)\log^4(LN)$$

with probability $1 - O(M(NL)^{-\beta})$. To complete the proof, we note that if we have

$$M \geq c\beta\kappa R\left(N + L\mu_0^2\right)\log^2(LN)$$

then both terms in this bound are less than μ_{k-1}^2, proving that the coherence is non-increasing over the coarse of the golfing scheme.

□

8.7.1 Matrix Bernstein Inequality and Olicz Norm

The majority of the proofs required to show out main result depend heavily on the matrix Bernstein inequality, as outlined in [243]. This inequality essentially utilizes the variance measure and Oricz norm of a matrix to bound the largest singular value of the matrix. The matrix Bernstein inequality is outlined as

Theorem 9 (Matrix Bernstein's Inequality)**.** *Let $\boldsymbol{X}_i \in \mathbb{R}^{L,N}$, $i \in [1,\ldots,M]$ be M random matrices such that $\mathcal{E}[\boldsymbol{X}_i] = 0$ and $\|\boldsymbol{X}_i\|_{\psi_\alpha} < U_\alpha < \infty$ for some $\alpha \geq 1$. Then with probability $1 - e^{-}t$, the spectral norm of the sum is bounded by*

$$\left\|\sum_{i=1}^{M} \boldsymbol{X}_i\right\| \leq C \max\left\{\sigma_X\sqrt{t + \log(L+N)}, U_\alpha \log^{1/\alpha}\left(\frac{MU_\alpha^2}{\sigma_X^2}\right)(t + \log(L+N))\right\}$$

for some constant C and the variance parameter defined by

$$\sigma_X = \max\left\{\left\|\sum_{i=1}^{M}\mathcal{E}[\boldsymbol{X}_i\boldsymbol{X}_i^*]\right\|^{1/2}, \left\|\sum_{i=1}^{M}\mathcal{E}[\boldsymbol{X}_i^*\boldsymbol{X}_i]\right\|^{1/2}\right\}$$

where the Orlicz-α norm $\|X\|_{\psi_\alpha}$ is defined as

$$\|X\|_{\psi_\alpha} = \inf\left\{y > 0 | \mathcal{E}\left[e^{\|X\|^\alpha/y^\alpha}\right] \leq 2\right\}$$

In particular we will utilize the matrix Bernstein inequality with the Orlicz-1 and Orlicz-2 norms, since subgaussian and subexponential random variables have bounded Orlicz-2 and -1 norms, respectively. To calculate these norms, we will find the following lemmas from [98, 243] useful:

Lemma 11 (Lemma 5.14 in [243])**.** *A random variable X is subgaussian iff X^2 is subexponential. Furthermore,*

$$\|X\|_{\psi_2}^2 \leq \left\|X^2\right\|_{\psi_1} \leq 2\|X\|_{\psi_2}^2 .$$

Lemma 12 (Lemma 7 in [98])**.** *Let X_1 and X_2 be two subgaussian ranfom variables. Then the product X_1X_2 is a subexponential random variable with*

$$\|X_1X_2\|_{\psi_1} \leq c\|X_1\|_{\psi_2}\|X_2\|_{\psi_2} .$$

Lemma 11 essentially relates the Orlicz-1 and -2 norms for a random variable and it's square. Lemma 12 allows us to factor an Orlicz-1 norm of a sub-exponential random variable when the random variable can be written as the product of two subgaussian random variables. Finally we find useful the following calculation for the Orlicz-1 norm of the norm of a random Gaussian vector $\boldsymbol{z}_n$ with *i.i.d.* zero-mean and variance σ^2 entries:

$$\begin{aligned}
\left\|\|\boldsymbol{z}_n\|_2^2\right\|_{\psi_1} &= \inf\left\{y : \mathcal{E}\left[e^{\|\boldsymbol{z}_n\|_2^2/y}\right] \leq 2\right\} \\
&= \inf\left\{y : \left(\mathcal{E}\left[e^{z_n^2/y}\right]\right)^M \leq 2\right\} \\
&= \inf\left\{y : \frac{1}{\sqrt{2\pi\sigma}}\int_{\mathbb{R}} e^{-z_n^2(1/2\sigma^2-1/y)}dz_n \leq 2^{\frac{1}{M}}\right\} \\
&= \inf\left\{y : \frac{\sqrt{2\pi\frac{y\sigma^2}{y-2\sigma^2}}}{\sqrt{2\pi}\sigma}\int_{\mathbb{R}} \frac{1}{\sqrt{2\pi\frac{y\sigma^2}{y-2\sigma^2}}} e^{-z_n^2(1/2\sigma^2-1/y)}dz_n \leq 2^{\frac{1}{M}}\right\} \\
&= \inf\left\{y : \sqrt{\frac{y}{y-2\sigma^2}} \leq 2^{\frac{1}{M}}\right\} \\
&= \inf\left\{y : y \geq 2\frac{4^{\frac{1}{M}}\sigma^2}{4^{\frac{1}{M}}-1}\right\} \\
&= \inf\left\{y : y \geq \frac{2\sigma^2}{1-4^{-\frac{1}{M}}}\right\} \\
&= \frac{2\sigma^2}{1-4^{-\frac{1}{M}}}
\end{aligned} \tag{121}$$

8.8 Derivation of recovery bound for infinite length inputs

In this appendix we derive the bound in Equation (34) of the main text. The approach we take is to bound the individual components of Equation (9) of the main text. As the noise term due to noise in the inputs is unaffected, we will bound the noise term due to the unrecovered signal (the first term in Equation (9) of the main text) by the component of the input history that is beyond the attempted recovery, and we will bound the signal approximation term (the second term in Equation (9) of the main text) by the quality of the signal recovery possible in the attempted recovery length. In this way we can observe how different properties of the system and input sequence affect signal recovery.

To bound the first term in Equation (9) of the main text (i.e., the omission errors due to inputs beyond the recovery window), we first write the current state at any time N^* as

$$\boldsymbol{x}[N^*] = \sum_{n=0}^{N^*} \boldsymbol{W}^{N^*-n} \boldsymbol{z} s[n].$$

We only wish to recover the past $N \leq N^*$ time steps, so we break up the summation into components of the current state due to "signal" (i.e., signal we attempt to recover) and "noise" (i..e, older signal we omit from the recovery):

$$\begin{aligned}\boldsymbol{x}[N^*] &= \sum_{n=N^*-N+1}^{N^*} \boldsymbol{W}^{N^*-n} \boldsymbol{z} s[n] + \sum_{n=0}^{N^*-N} \boldsymbol{W}^{N^*-n} \boldsymbol{z} s[n] \\ &= \sum_{n=N^*-N+1}^{N^*} \boldsymbol{W}^{N^*-n} \boldsymbol{z} s[n] + \boldsymbol{\epsilon} \\ &= \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{\epsilon}_2.\end{aligned}$$

From here we can see that the first summation is the matrix multiply $\boldsymbol{\Phi}\boldsymbol{x}$ as is discussed in the paper. The second summation here, $\boldsymbol{\epsilon}_2$, essentially acts as an additional noise term in the recovery. We can further analyze the effect of this noise term by understanding that $\boldsymbol{\epsilon}_2$ is bounded for well behaved input sequences $s[n]$ (in fact all that is needed is that the maximum value or the expected value and variance are reasonably bounded) when the eigenvalues of $\boldsymbol{W}$ are of magnitude $q \leq 1$. We can explicitly calculate the worst case

scenario bounds on the norm of $\boldsymbol{\epsilon}_2$,

$$\begin{aligned}\left\|\sum_{n=0}^{N^*-N} \boldsymbol{W}^{N^*-n}\boldsymbol{z}s[n]\right\|_2 &\leq \left\|\sum_{n=0}^{N^*-N} \boldsymbol{U}\left(q\boldsymbol{D}\right)^{N^*-n}\boldsymbol{U}^{-1}\boldsymbol{z}s[n]\right\|_2 \\ &\leq \|\boldsymbol{U}\|_2\left\|\sum_{n=0}^{N^*-N}\left(q\boldsymbol{D}\right)^{N^*-n}\boldsymbol{U}^{-1}\boldsymbol{z}s[n]\right\|_2,\end{aligned}$$

where $\boldsymbol{D} = \text{diag}(d_1, \ldots, d_M)$ is the diagonal matrix containing the normalized eigenvalues of $\boldsymbol{W}$. If we assume that $\boldsymbol{z}$ is chosen as mentioned in Section 3.1.2 so that $\boldsymbol{U}^{-1}\boldsymbol{z} = \left(1/\sqrt{M}\right)\mathbf{1}$, the eigenvalues of $\boldsymbol{W}$ are uniformly spread around a complex circle of radius q, and that $s[n] \leq s_{\max}$ for all n, then we can bound this quantity as

$$\begin{aligned}\left\|\sum_{n=0}^{N^*-N} \boldsymbol{W}^{N^*-n}\boldsymbol{z}s[n]\right\|_2 &\leq \frac{s_{\max}}{\sqrt{M}}\|\boldsymbol{U}\|_2\left\|\sum_{n=0}^{N^*-N}\left(q\boldsymbol{D}\right)^{N^*-n}\mathbf{1}\right\|_2 \\ &= \frac{s_{\max}}{\sqrt{M}}\|\boldsymbol{U}\|_2\left\|\begin{bmatrix}\sum_{n=0}^{N^*-N} q^{N^*-n}d_1^{N^*-n} \\ \vdots \\ \sum_{n=0}^{N^*-N} q^{N^*-n}d_M^{N^*-n}\end{bmatrix}\right\|_2 \\ &= \frac{s_{\max}}{\sqrt{M}}\|\boldsymbol{U}\|_2\sqrt{\sum_{k=1}^{M}\left|\sum_{n=0}^{N^*-N} q^{N^*-n}d_k^{N^*-n}\right|^2} \\ &\leq s_{\max}\|\boldsymbol{U}\|_2\left|\sum_{n=0}^{N^*-N} q^{N^*-n}\right| \\ &\leq s_{\max}\|\boldsymbol{U}\|_2\left|\frac{q^N - q^{N^*}}{1-q}\right|\end{aligned}$$

where d_k is the k^{th} normalized eigenvalue of $\boldsymbol{W}$. In the limit of large input signal lengths ($N^* \to \infty$), we have $N^* \gg N$ and so $q^N \gg q^{N^*}$, which leaves the approximate expression

$$\|\boldsymbol{\epsilon}_2\|_2 \leq s_{\max}\|\boldsymbol{U}\|_2\left|\frac{q^N}{1-q}\right|.$$

To bound the second term in Equation (9) (i.e., the signal approximation errors due to imperfect recovery), we must characterize the possible error between the signal (which is S-sparse) and the approximation to the signal with the K^* largest coefficients. In the worst case scenario, there are $S - K^* + 1$ coefficients that cannot be guaranteed to be recovered

by the RIP conditions, and these coefficients all take the maximum value $s_{\max}$. In this case, we can bound the signal approximation error as stated in the main text:

$$\begin{aligned}\frac{\beta}{\sqrt{K^*}}\left\|\boldsymbol{s}-\boldsymbol{s}_{S^*}\right\|_1 &\leq \frac{\beta}{\sqrt{K^*}}\sum_{n=K^*+1}^{S}\left|q^n s_{\max}\right| \\ &= \frac{\beta s_{\max}}{\sqrt{K^*}}\left(\frac{q^{K^*}-q^S}{1-q}\right).\end{aligned}$$

In the case where noise is present, we can also bound the total power of the noise term,

$$\alpha\left\|\sum_{k=0}^{N+N^*}\boldsymbol{W}^k\widetilde{\boldsymbol{z\epsilon}}[k]\right\|_2^2,$$

using similar steps. Taking $\epsilon_{\max}$ as the largest possible input noise into the system, we obtain the bound

$$\alpha\left\|\sum_{k=0}^{N+N^*}\boldsymbol{W}^k\widetilde{\boldsymbol{z\epsilon}}[k]\right\|_2^2 < \alpha\epsilon_{\max}\left\|\boldsymbol{U}\right\|_2\left|\frac{q}{1-q}\right|$$

REFERENCES

[1] A. Charles, H. Yap, and C. Rozell, "Short term network memory capacity via the restricted isometry property," *Neural Computation*, vol. 26, June 2014.

[2] A. Charles, D. Yin, and C. Rozell, "Retention of multiple memory streams in random neural networks." In Prep, 2015.

[3] H. L. Yap, A. S. Charles, and C. J. Rozell, "The restricted isometry property for echo state networks with application to sequential memory capacity," *Proceedings of the Statistical Signal Processing Workshop, Ann Arbor, Michigan*, August 2012.

[4] A. S. Charles, D. Yin, and C. J. Rozell, "Can random linear networks store multiple long input streams?," *Proceedings of the IEEE GlobalSIP*, Dec 2014.

[5] A. S. Charles, H. L. Yap, and C. J. Rozell, "Using compressed sensing to study sequence memory capacity in networked systems," *SPARS13*, January 2013.

[6] A. Charles, H. Yap, , and C. Rozell, "Short term memory in neural networks via the restricted isometry property," *Computational Neuroscience Meeting Workshop on Methods of Information Theory in Computational Neuroscience, Atlanta, GA*, July 2012.

[7] A. S. Charles, H. L. Yap, and C. J. Rozell, "Short-term memory capacity in recurrent networks via compressed sensing," *Janelia Farm Conference on Machine Learning, Statistical Inference, and Neuroscience, Ashburn, Virginia*, May 2012.

[8] H. L. Yap, A. S. Charles, and C. J. Rozell, "Short-term memory capacity in recurrent networks via compressed sensing," *Challenges in Geometry, Analysis and Computation: High-Dimensional Synthesis, Yale University*, June 2012.

[9] A. S. Charles, H. L. Yap, and C. J. Rozell, "Short-term memory capacity in recurrent neural networks via compressive sensing," *Presented at the COSYNE 2012 workshop, Salt Lake City, Utah*, February 2012.

[10] A. Charles and C. Rozell, "Re-weighted dynamic filtering for time-varying sparse signal estimation." Submitted, August 2014.

[11] A. Charles and C. Rozell, "Learning bilinear dynamics models in heirarchical sparsity-based dynamic filtering." In Prep, 2015.

[12] A. S. Charles and C. J. Rozell, "Convergence of basis pursuit de-noising with dynamic filtering," *Proceedings the IEEE GlobalSIP*, Dec 2014.

[13] A. S. Charles and C. J. Rozell, "Dynamic filtering of sparse signals using reweighted ℓ_1," *Proceedings of ICASSP*, May 2013.

[14] M. S. Asif, A. S. Charles, J. Romberg, and C. J. Rozell, "Estimation and dynamic updating of time-varying signals with sparse variations," *Proceedings of the ICASSP, Prague, Czech Republic*, May 2011.

[15] A. S. Charles, M. S. Asif, J. Romberg, and C. J. Rozell, "Sparsity penalties in dynamical system estimation," *Proceedings of CISS, Baltimore, Maryland*, March 2011.

[16] A. S. Charles and C. J. Rozell, "Robust estimation of sparse time-varying signals," *ITA 2015, La Jolla, California*, Feb 2015.

[17] A. S. Charles and C. J. Rozell, "Stochastic filtering via reweighted ℓ_1," *SPARS13*, January 2013.

[18] C. Rozell and A. Charles, "Recursive estimation of dynamic signals with sparsity models via re-weighted l1 minimization," *Janelia Farm Conference on Machine Learning, Statistical Inference, and Neuroscience, Ashburn, VA*, May 2012.

[19] A. S. Charles and C. J. Rozell, "A hierarchical re-weighted-ℓ1 approach for dynamic sparse signal estimation," *SPARS11 workshop, Edinburgh, Scotland UK*, June 2011.

[20] A. S. Charles, B. A. Olshausen, and C. J. Rozell, "Learning sparse codes for hyperspectral imagery," *IEEE Jour. of Sel. Top. in Sig. Proc.*, vol. 5, no. 5, pp. 963–978, 2011.

[21] A. Charles and C. Rozell, "Spectral super-resolution of hyperspectral imagery using reweighted ℓ_1 spatial filtering," *IEEE G. and Remote Sens. Lett.*, 2013. Accepted.

[22] A. S. Charles, B. A. Olshausen, and C. J. Rozell, "Sparse coding for spectral signatures in hyperspectral images," *Proceedings of the Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA*, November 2010.

[23] A. S. Charles, C. J. Rozell, and N. Tufillaro, "Sparsity based spectral super-resolution and applications to ocean water color," *IGARSS, Québec, Canada*, May 2014. Invited contribution.

[24] C. Rozell and A. Charles, "Spectral super-resolution of hyperspectral images," *SIAM Conference on Imaging Science, Philadelphia, PA*, May 2012.

[25] A. S. Charles, B. Olshausen, and C. J. Rozell, "Learning sparse codes for hyperspectral images," *Duke Workshop on Sensing and Analysis of High-Dimensional Data (SAHD), Durham, NC*, July 2011.

[26] A. S. Charles, P. Garrigues, and C. J. Rozell, "A common network architecture efficiently implements a variety of sparsity-based inference problems," *Neural Comp.*, vol. 24, no. 12, pp. 3317–3339, 2012.

[27] S. Shapero, A. S. Charles, C. Rozell, and P. Hasler, "Low power sparse approximation on reconfgurable analog hardware," *IEEE Jour. on Emer. and Sel. Top. in Circ. and Sys.*, vol. 2, no. 3, pp. 530–541, 2011.

[28] A. S. Charles, A. A. Kressner, and C. J. Rozell, "Causal sparse decompositions of audio signals," *Proceedings of the IEEE Signal Processing (DSP) Workshop, Sedona, AZ*, January 2011. (Nominated for best student paper).

[29] C. J. Rozell, M. Zhu, A. S. Charles, H. L. Yap, and M. Norko, "The role of sparsity in visual perception," *BICA*, Nov 2014.

[30] A. Kressner, A. Charles, and C. Rozell, "Causal locally competitive algorithm for the sparse decomposition of audio signals," *IEEE Womens Workshop on Communications and Signal Processing, Ban, Canada*, July 2012.

[31] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[32] B. A. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, Jun 1996.

[33] M. Aharon, M. Elad, A. Bruckstein, and Y. Katz, "K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representations," *IEEE Proceedings - Special Issue on Applications of Compressive Sensing & Sparse Representation*, 2006.

[34] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large scale l1-regularized least squares," *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, pp. 606–617, Dec 2007.

[35] C. J. Rozell, D. H. Johnson, and R. G. B. B. A. Olshausen, "Sparse coding via thresholding and local competition in neural circuits," *Neur. Comp.*, vol. 20, no. 10, pp. 2526–2563, 2008.

[36] S. Shapero, C. Rozell, and P. Hasler, "Configurable hardware integrate and fire neurons for sparse approximation," *Neural Networks*, vol. 45, pp. 134–143, 2013.

[37] M. Zhu and C. J. Rozell, "Visual nonclassical receptive field effects emerge from sparse coding in a dynamical system," *PLoS computational biology*, vol. 9, no. 8, p. e1003191, 2013.

[38] T. Hu, A. Genkin, and D. B. Chklovskii, "A network of spiking neurons for computing sparse representations in an energy-efficient way," *Neural computation*, vol. 24, no. 11, pp. 2852–2872, 2012.

[39] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1631–1642, 2013.

[40] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[41] W. Maass, T. Natschlager, and H. Markram, "A model for real-time computation in generic neural microcircuits," *Advances in neural information processing systems*, pp. 229–236, 2003.

[42] D. Verstraeten, B. Schrauwen, D. Stroobandt, and J. Van Campenhout, "Isolated word recognition with the liquid state machine: a case study," *Information Processing Letters*, vol. 95, no. 6, pp. 521–528, 2005.

[43] F. Schürmann, K. Meier, and J. Schemmel, "Edge of chaos computation in mixed-mode vlsi-a hard liquid," in *Advances in Neural Information Processing Systems*, pp. 1201–1208, 2004.

[44] W. Maass, R. Legenstein, and H. Markram, "A new approach towards vision suggested by biologically realistic neural microcircuit models," in *Biologically Motivated Computer Vision*, pp. 282–293, 2002.

[45] H. Burgsteiner, "On learning with recurrent spiking neural networks and their applications to robot control with real-world devices," *PhD Thesis, Graz University of Technology*, 2005.

[46] H. Jaeger, "Reservoir riddles: Suggestions for echo state network research," in *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, vol. 3, pp. 1460–1462, 2005.

[47] J. Hertzberg, H. Jaeger, and F. Schönherr, "Learning to ground fact symbols in behavior-based robots," in *ECAI*, pp. 708–712, 2002.

[48] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.

[49] R. Baraniuk, V. Cevher, and M. Wakin, "Low-dimensional models for dimensionality reduction and signal recovery: A geometric perspective," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 959–971, 2010.

[50] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91.

[51] E. J. C. T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[52] E. J. Candes, "Compressive sampling," *Proc. Int. Congr. Mathematicians*, vol. 3, pp. 1433–1452, 2006.

[53] H. R. H, "Compressive sensing and structured random matrices," *Theoretical Found. and Numerical Methods for Sparse Recovery*, vol. 9, pp. 1–92, 2010.

[54] D. L. Donoho and J. Tanner, "Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 367, no. 1906, pp. 4273–4293, 2009.

[55] M. Rudelson and R. Vershynin, "On sparse reconstruction from fourier and gaussian measurements," *Communications on Pure and Applied Mathematics*, vol. 61, no. 8, pp. 1025–1045, 2008.

[56] R. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, pp. 118–121, Jul 2007.

[57] B. A. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[58] D. Needell and J. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[59] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Asilomar Conference on Signals, Systems and Computers*, pp. 40–44, 1993.

[60] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *Information Theory, IEEE Transactions on*, vol. 53, no. 12, pp. 4655–4666, 2007.

[61] L. Rebollo-Neira and D. Lowe, "Optimized orthogonal matching pursuit approach," *Signal Processing Letters, IEEE*, vol. 9, no. 4, pp. 137–140, 2002.

[62] E. Candes and T. Tao, "The dantzig selector: Statistical estimation when p is much larger than n," *The Annals of Statistics*, pp. 2313–2351, 2007.

[63] E. J. Candes and Y. Plan, "A probabilistic and ripless theory of compressed sensing," *Information Theory, IEEE Transactions on*, vol. 57, no. 11, pp. 7235–7254, 2011.

[64] P. Garrigues and B. Olshausen, "Group sparse coding with a laplacian scale mixture prior," *Advances in Neural Information Processing Systems*, vol. 24, 2010.

[65] M. Wainwright and M. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, no. 1-2, pp. 1–305, 2008.

[66] D. R. Smith, *Variational methods in optimization*. Courier Dover Publications, 1998.

[67] M. M. Denn, *Optimization by variational methods*. McGraw-Hill, 1969.

[68] R. Little and D. Rubin, *Statistical Analysis with Missing Data*. John Wiley & Sons, New York, 1987.

[69] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal processing magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[70] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*, vol. 382. John Wiley & Sons, 2007.

[71] Y. Karklin and M. S. Lewicki, "A hierarchical bayesian model for learning nonlinear statistical regularities in nonstationary natural signals," *Neural Computation*, vol. 17, no. 2, pp. 397–423, 2005.

[72] Y. Karklin and M. S. Lewicki, "Learning higher-order structures in natural images," *Network: Computation in Neural Systems*, vol. 14, no. 3, pp. 483–499, 2003.

[73] B. J. Culpepper and B. A. Olshausen, "Learning transport operators for image manifolds," *Advances in Neural Information Processing Systems*, vol. 22, 2010.

[74] J. Romberg, "Compressive sensing by random convolution," *SIAM Journal on Imaging Sciences*, vol. 2, no. 4, pp. 1098–1128, 2009.

[75] H. Jaeger, "Short term memory in echo state networks," *GMD Report 152 German National Research Center for Information Technology*, 2001.

[76] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[77] D. Verstraeten, J. Dambre, X. Dutoit, and B. Schrauwen, "Memory versus nonlinearity in reservoirs," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2010.

[78] T. Strauss, W. Wustlich, and R. Labahn, "Design strategies for weight matrices of echo state networks," *Neural Computation*, vol. 24, no. 12, pp. 3246–3276, 2012.

[79] E. Wallace, R. M. Hamid, and P. E. Latham, "Randomly connected networks have short temporal memory," *Neural Computation*, vol. 25, pp. 1408–1439, 2013.

[80] J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, "Information processing capacity of dynamical systems," *Scientific reports*, vol. 2, 2012.

[81] D. Verstraeten, B. Schrauwen, M. dHaene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural Networks*, vol. 20, no. 3, pp. 391–403, 2007.

[82] M. LukošEvičIus and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.

[83] S. Ganguli, D. Huh, and H. Sompolinsky, "Memory traces in dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 105, no. 48, p. 18970, 2008.

[84] S. Ganguli and H. Sompolinsky, "Short-term memory in neuronal networks through dynamical compressed sensing," *Conf. on Neural Info. Proc. Sys.*, 2010.

[85] O. White, D. Lee, and H. Sompolinsky, "Short-term memory in orthogonal neural networks," *Physical review letters*, vol. 92, no. 14, p. 148102, 2004.

[86] S. Becker, E. J. Candes, and M. Grant, "Templates for convex cone problems with applications to sparse signal recovery," *Mathematical Programming Computation*, vol. 3, August 2011.

[87] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2053–2080, 2010.

[88] E. Candes and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, 2010.

[89] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.

[90] P. Chen and D. Suter, "Recovering the missing components in a large noisy low-rank matrix: Application to sfm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1051–1063, 2004.

[91] M. Fazel, "Matrix rank minimization with applications," *PhD Thesis, Stanford University*, 2002.

[92] A. Singer and M. Cucuringu, "Uniqueness of low-rank matrix completion by rigidity theory," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 4, pp. 1621–1641, 2010.

[93] C. Beck and R. D'Andrea, "Computational study and comparisons of lft reducibility methods," in *Proceedings of the American Control Conference*, vol. 2, pp. 1013–1017, 1998.

[94] M. Mesbahi and G. P. Papavassilopoulos, "On the rank minimization problem over a positive semidefinite linear matrix inequality," *IEEE Transactions on Automatic Control*, vol. 42, no. 2, pp. 239–243, 1997.

[95] K.-C. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems," *Pacific Journal of Optimization*, vol. 6, no. 615-640, p. 15, 2010.

[96] Z. Liu and L. Vandenberghe, "Interior-point method for nuclear norm approximation with application to system identification," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 3, pp. 1235–1256, 2009.

[97] M. Jaggi, M. Sulovsk, *et al.*, "A simple algorithm for nuclear norm regularized problems," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 471–478, 2010.

[98] A. Ahmed and J. Romberg, "Compressive multiplexing of correlated signals," *arXiv preprint arXiv:1308.5146*, 2013.

[99] P. Diaconis and M. Shahshahani, "On the eigenvalues of random matrices," *Journal of Applied Probability*, vol. 31, pp. 49–62, 1994.

[100] G. Mongillo, O. Barak, and M. Tsodyks, "Synaptic theory of working memory," *Science*, vol. 319, pp. 1543–1546., 2008.

[101] H. Jaeger and H. Haas, "Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.

[102] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk, "Signal processing with compressive measurements," *IEEE J. Sel. Topics Signal Proc.*, vol. 4, no. 2, pp. 445–460, 2010.

[103] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME-Journal of Basic Engineering*, vol. 82, no. D, pp. 35–45, 1960.

[104] S. Haykin, "Kalman filters," in *Kalman Filtering and Neural Networks* (S. Haykin, ed.), pp. 1–22, John Wiley & Sons, Inc., 2001.

[105] E. A. Wan and R. van der Merwe, "The unscented Kalman filter," in *Kalman Filtering and Neural Networks* (S. Haykin, ed.), pp. 221–282, John Wiley & Sons, Inc., 2001.

[106] D. Angelosante, S. I. Roumeliotis, and G. B. Giannakis, "Lasso-Kalman smoother for tracking sparse signals," *Proc of the Asilomar Conf. on Sig., Sys. and Comp.*, 2009.

[107] S. Farahmand, G. Giannakis, and D. Angelosante, "Doubly robust smoothing of dynamical processes via outlier sparsity constraints," *IEEE Transactions on Signal Processing*, vol. 59, pp. 4529–4543, Oct 2011.

[108] Z. Zhang and B. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning," *Selected Topics in Signal Processing, IEEE Journal of*, no. 99, pp. 1–1, 2011.

[109] A. Sankaranarayanan, P. Turaga, R. Baraniuk, and R. Chellappa, "Compressive acquisition of dynamic scenes," *Computer Vision–ECCV 2010*, pp. 129–142, 2010.

[110] M. S. Asif, L. Hamilton, M. Brummer, and J. Romberg, "Motion-adaptive spatio-temporal regularization (master) for accelerated dynamic mri," *Magnetic Resonance in Medicine*, vol. 70, no. 3, pp. 800–812, 2013.

[111] A. Carmi, P. Gurfil, and D. Kanevsky, "Methods for sparse signal recovery using Kalman filtering pseudo-measuremennt norms and quasi-norms," *IEEE Transactions on Signal Processing*, vol. 58, pp. 2405–2409, Apr 2010.

[112] N. Vaswani, "Kalman filtered compressed sensing," *Proc of ICIP 2008*, pp. 893–896, 2008.

[113] C. Qiu and N. Vaswani, "Recursive sparse recovery in large but low dimensional noise," *48th Allerton Conference on Communication Control and Computing*, 2011.

[114] N. Vaswani, "Ls-cs-residual (ls-cs): Compressive sensing on the lease squares residual," *IEEE Trans on Signal Processing*, vol. 58, pp. 4108–4120, Aug 2010.

[115] N. Vaswani and W. Lu, "Modified-cs: Modifying compressive sensing for problems with partially known support," *IEEE Trans. on Sig. Proc.*, vol. 58, no. 9, pp. 4595–4607, 2010.

[116] D. Sejdinovic, C. Andrieu, and R. Piechocki, "Bayesian sequential compressed sensing in sparse dynamical systems," in *Proc Allerton Conf. on Com., Cont., and Comp.*, pp. 1730–1736, IEEE, 2010.

[117] D. Zachariah, S. Chatterjee, and M. Jansson, "Dynamic iterative pursuit," *IEEE Trans. on Sig. Proc.*, 2012. In Press.

[118] J. Ziniel, L. C. Potter, and P. Schniter, "Tracking and smoothing of time-varying sparse signals via approximate belief propagation," *Proceesings of the Asilomar Conference on Signals, Systems and Computers*, 2010.

[119] M. A. Khajehnejad, W. Xu, S. Avestimehr, and B. Hassibi, "Weighted ℓ_1 minimization for sparse recovery with prior information," *http://arxiv.org/abs/0901.2912v1*, 2009.

[120] C. Miosso, R. von Borries, M. Argaez, L. Velazquez, C. Quintero, and C. M. Potes, "Compressive sensing reconstruction with prior information by iteratively reweighted least-squares," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2424–2431, 2009.

[121] R. E. Carrillo, L. F. Polania, and K. E. Barner, "Iterative algorithms for compressed sensing with partially known support," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pp. 3654–3657, 2010.

[122] J. Mattingley and S. Boyd, "Real-time convex optimization in signal processing," *IEEE Signal Processing Magazine*, vol. 27, pp. 50–61, May 2010.

[123] R. Baraniuk and P. Steeghs, "Compressive radar imaging," in *IEEE Radar Conference*, pp. 128–133, 2007.

[124] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on pure and applied mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[125] A. Balavoine, C. J. Rozell, and J. Romberg, "Discrete and continuous-time soft-thresholding with dynamic inputs," *arXiv preprint arXiv:1405.1361*, 2014.

[126] S. Chrétien and A. O. H. III, "Kullback proximal algorithms for maximum-likelihood estimation," *IEEE Transactions on Information Theory*, vol. 46, no. 5, pp. 1800–1810, 2000.

[127] D. Needell, "Noisy signal recovery via iterative reweighted l1-minimization," in *Forty-Third Asilomar Conference on Signals, Systems and Computers*, pp. 113–117, 2009.

[128] C. F. Cadieu and B. A. Olshausen, "Learning intermediate-level representations of form and motion from natural movies," *Neur. Comp.*, vol. 24, no. 4, pp. 827–866, 2012.

[129] J. Ziniel and P. Schniter, "Dynamic compressive sensing of time-varying signals via approximate message passing," *IEEE Tans on Sig Proc*, vol. 61, pp. 5270–5284, July 2013.

[130] I. Selesnick, R. Baraniuk, and N. Kingsbury, "The dual-tree complex wavelet transform," *IEEE Sig. Proc. Mag.*, vol. 22, no. 6, pp. 123–151, 2005.

[131] R. Coifman, F. Geshwind, and Y. Meyer, "Noiselets," *Applied Computational Harmonic Analysis*, vol. 10, no. 1, pp. 27–44, 2001.

[132] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 14–20, 2008.

[133] S. Becker, E. Candès, and M. Grant, "Templates for convex cone problems with applications to sparse signal recovery," *Mathematical Programming Computation*, pp. 1–54, 2011.

[134] "The benefits of the 8-spectral bands of worldview-2," Mar 2010. Available Online at `http://www.digitalglobe.com/downloads/spacecraft/WorldView-2_8-Band_Applications_Whitepaper.pdf`.

[135] J. P. Kerkes and J. R. Schott, "Hyperspectral imaging systems," in *Hyperspectral Data Exploitation: Theory and Applications* (C.-I. Chang, ed.), pp. 19–45, John Wiley & Sons, Inc., 2007.

[136] C. M. Bachmann, T. L. Ainsworth, and R. A. Fusina, "Exploiting manifold geometry in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 441–454, 2005.

[137] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Proceesing Magazine*, vol. 19, pp. 44–57, Jan 2002.

[138] A. Plaza, P. Martinez, R. Perez, and J. Plaza, "Spatial/spectral endmember extraction by multidimensional morphological operations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, pp. 2025–2041, Sep 2002.

[139] M. Winter, "N-FINDR: An algorithm for fast autonomous spectral end-member determination in hyperspectral data," *IEEE Porceedings of SPIE*, vol. 3753, no. 5, 1999.

[140] A. Bateson and B. Curtiss, "A method for manual endmember selection and spectral unmixing," *Remote Sens. Environ.*, vol. 55, pp. 229–243, Mar 1996.

[141] J. Bowles, P. J. Palmadesso, J. A. Antoniadas, M. M. Baumback, and J. L. Rickard, "Use of filter vectors in hyperspectral data analysis," *Proceedings of SPIE, Infrared Spaceborne Remote Sensing III*, pp. 148–157, 1995.

[142] A. Ifarraguerri and C.-I. Chang, "Multispectral and hyperspectral image analysis with convex cones," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, pp. 756–770, Mar 1999.

[143] C. Gomez, H. L. Borgne, P. Allemand, C. Delacourt, and P. Ledru, "N-FindR method versus independent component analysis for lithological identification in hyperspectral imagery," *International Journal of Remote Sensing*, vol. 28, pp. 5315–5338, Jun 2007.

[144] O. Forni, F. Poulet, J.-P. Bibring, S. Erard, C. Gomez, Y. Langevin, B. Gondet, and Omega Team, "Component separation of omega spectra with ica," *Lunar and Planetary Science Technical Report*, vol. 1623, 2005.

[145] S. Erard, P. Drossart, and G. Piccioni, "Multivariate analysis of visible and infrared thermal imaging spectrometer (virtis) venus express nightside and limb observations," *Journal of Geophysical Research*, vol. 114, pp. 1–20, Jan 2009.

[146] M. Berman, H. Kiiveri, R. Lagerstrom, A. Ernst, R. Dunne, and J. F. Huntington, "ICE: A statistical approach to itentifying endmembers in hyperspectral images," *IEEE Transaction on Geoscience and Remote Sensing*, vol. 42, pp. 2085–2095, Oct 2004.

[147] D. Rogge, B. Rivard, J. Zhang, and J. Feng, "Iterative spectral unmixing for optimizing per-pixel endmember sets," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, Dec 2006.

[148] M. Elad, M. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *IEEE Proceedings - Special Issue on Applications of Compressive Sensing & Sparse Representation*, Oct 2008.

[149] Z. Guo, T. Wittman, and S. Osher, "L1 unmixing and its application to hyperspectral image enhancement," *Proc. SPIE Conference on Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV*, Apr 2009. Orlando, Florida.

[150] M.-D. Iordache, J. M. B. Dias, and A. Plaza, "Sparse unmixing of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, 2010.

[151] J. Greer, "Sparse demixing," *IEEE Proceedings of SPIE*, vol. 7695, pp. 76951O–76951O–12, May 2010.

[152] B. A. Olshausen and D. J. Field, "Sparse coding of sensory inputs," *Current Opinion in Neurobiology*, vol. 14, pp. 481–487, 2004.

[153] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

[154] S. Osher, B. D. Y. Mao, and W. Yin, "Fast linearized bregman iteration for compressive sensing and sparse denoising," *Adv. in Neur. Info. Proc. Sys.*, pp. 505–512, 2008.

[155] J. Bioucas-Dias and M. Figueiredo, "Alternating direction algorithms for constrained sparse regression application to hyperspectral unmixing," *2nd IEEE GRSS Workshop on Hyperspectral Image and Signal Processing -WHISPERS'2010*, 2010. Raykjavik, Iceland.

[156] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, D. Dunson, G. Sapiro, and L. Carin, "Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images," *IMA Preprint Series #2307*, Apr 2010.

[157] M. Aharon, M. Elad, and A. Bruckstein, "K-svd and its non-negative variant for dictionary design," *Proceedings of the SPIE conference on wavelets*, vol. 5914, Jul 2005.

[158] A. M. Bruckstein, M. Elad, and M. Zibulevsky, "On the uniqueness of nonegative sparse solutions to underdetermined sparse solutions to underdetermined systems of equations," *IEEE Transactions on Information Theory*, vol. 54, pp. 4813–4820, Nov 2008.

[159] Q. Geng, H. Wang, and J. Wright, "On the local correctness of ℓ^1-minimization for dictionary learning," *arXiv preprint arXiv:1101.5672v1*, 2011. http://arxiv4.library.cornell.edu/pdf/1101.5672v1.

[160] C. M. Bachmann, T. F. Donato, G. M. Lamela, W. J. Rhea, M. H. Bettenhausen, R. A. Fusina, K. R. D. Bois, J. H. Porter, and B. R. Truitt, "Automatic classification of land cover on smith island, va, using hymap imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, pp. 2313–2330, Oct 2002.

[161] C. M. Bachmann, "Improving the performance of classifiers in high-dimensional remote sensing applications: An adaptive resampling strategy for error-prone exemplars (ARESEPE)," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, pp. 2101–2112, Sept 2003.

[162] C. M. Bachmann, M. H. Bettenhausen, R. A. Fusina, T. F. Donato, A. L. Russ, J. W. Burke, G. M. Lamela, W. J. Rhea, and B. R. Truitt, "A credit assignment approach to fusing classifiers of multiseasonal hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, pp. 2488–2499, Nov 2003.

[163] C. M. Bachmann, T. F. Donato, G. Lamela, J. Rhea, M. Bettenhausen, R. A. Fusina, K. D. Bois, J. Porter, and B. Truitt, "Automatic classification of land cover on Smith Island, VA, using HyMAP imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 10, pp. 2313–2330, 2002.

[164] O. Forni, S. M. Clegg, R. C. Wiens, and S. M. anc O. Gasnault, "Multivariate analysis of ChemCam first calibration samples," *40th Lunar and Planetary Science Conference*, vol. 1523, 2009.

[165] T. Wachtler, T. Lee, and T. J. Sejnowski, "Chromatic structure od natural scenes," *Journal of the Optical Society of America*, vol. 18, pp. 65–77, Jan 2001.

[166] S. Moussaoui, H. Hauksdottir, F. Schmidt, C. Jutten, J. Chanussot, D. Brie, S. Doute, and J. A. Benediksson, "On the decomposition of mars hyperspectral data by ica and bayesian positive source separation," *Neurocomputing*, vol. 71, pp. 2194–2208, Jun 2008.

[167] S. Jia and Y. Qian, "Constrained nonegative matrix factorization for hyperspectral imaging," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, pp. 161–173, Jan 2009.

[168] V. P. Pauca, J. Piper, and R. J. Plemmons, "Nonnegative matrix factorization for spectral data analysis," *Journal of Linear Algebra and its Applications*, vol. 416, no. 1, pp. 29–47, 2006.

[169] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, Nov 2004.

[170] A. Castrodad, Z. Xing, J. Greer, E. Bosch, L. Carin, and G. Sapiro, "Discriminative sparse representations in hyperspectral imagery," *IMA Preprint Series #2319*, Mar 2010.

[171] M. E. Gehm, R. John, D. J. Brady, R. M. Willett, and T. J. Schulz, "Single-shot compressive spectral imaging with dual-disperser architecture," *Optics Express*, vol. 15, pp. 14013–14026, Oct 2007.

[172] A. Wagadarikar, R. John, R. Willett, and D. Brady, "Single disperser design for coded aperture snapshot spectral imaging," *Computational Optical Sensing and Imaging, Applied Optics*, vol. 47, pp. B44–B51, Apr 2008.

[173] D. Manolakis and G. Shaw, "Detection algorithms for hyperspectral imaging applications," *IEEE Proceesings on Signal Processing*, vol. 19, pp. 29–43, Jan 2002.

[174] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," *IEEE Proceedings International Conference on Computer Vision (ICCV'09)*, 2009.

[175] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng, "Self-taught learning: Transfer learning from unlabled data," *Proceedings of the 24th International Conference on Machine Learning*, pp. 759–766, 2007.

[176] K. L. Oehler and R. M. Gray, "Combining image compression and classification using vector quantization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, May 1995.

[177] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[178] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[179] A. Y. Yang, J. Wright, Y. Ma, and S. Sastry, "Feature selection in face recognition:a sparse representation perspective," Tech. Rep. UCB/EECS-2007-99, University of California, Berkeley, 2007.

[180] E. Candès, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, 2008.

[181] D. Wipf and S. Nagarajan, "Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 317–329, 2010.

[182] A. Castrodad, Z. Xing, J. Greer, E. Bosch, L. Carin, and G. Sapiro, "Learning discriminative sparse representations for modeling, source separation, and mapping of hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4263–4281, 2011.

[183] A. P. Crosta, C. Sabine, and J. V. Taranik, "Hydrothermal alteration mapping at bodie, california, using aviris hyperspectral data," *Remote Sensing of Environment*, vol. 65, no. 3, pp. 309–319, 1998.

[184] E. J. Candes and J. Romberg, "ℓ_1-Magic: Recovery of sparse signals via convex programming," *http://www.acm.caltech.edu/l1magic/*, 2005.

[185] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing*, 2007.

[186] D. M. Malioutov, M. Cetin, and A. S. Willsky, "Homotopy continuation for sparse signal representation," *IEEE Proceedings of ICASSP*, 2005.

[187] M. S. Asif and J. Romberg, "Dynamic updating for ℓ_1 minimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 421–434, Apr 2010.

[188] P. Garrigues and L. Ghaoui, "An homotopy algorithm for the lasso with online observations," in *Neural Information Processing Systems*, vol. 21, pp. 489–496, 2008.

[189] J. Bradley, A. Kyrola, D. Bickson, and C. Guestrin, "Parallel coordinate descent for l1-regularized loss minimization," *Arxiv preprint arXiv:1105.5379*, 2011.

[190] S. Lee and S. Wright, "Implementing algorithms for signal and image reconstruction on graphical processing units," *Computer Sciences Department, University of Wisconsin-Madison, Tech. Rep*, 2008.

[191] T. Blumensath, M. Yaghoobi, and M. Davies, "Iterative hard thresholding and l0 regularisation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 877–880, 2007.

[192] J. M. Bioucas-Dias and M. A. T. Figueiredo, "A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2992–3004, 2007.

[193] E. T. Hale, W. Yin, and Y. Zhang, "A fixed-point continuation method for ℓ_1-regularized minimization with applications to compressed sensing," tech. rep., Rice University Department of Computational and Applied Mathematics, Jul 2007.

[194] M. Figueiredo and R. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. on Image Proc.*, vol. 12, no. 8, pp. 906–916, 2003.

[195] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.

[196] A. Balavoine, J. Romberg, and C. Rozell, "Convergence and rate analysis of neural networks for sparse approximation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 1377–1389, Sept 2012.

[197] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, p. 2554, 1982.

[198] S. Shapero, C. J. Rozell, and P. Hasler, "Configurable hardware integrate and fire neurons for sparse approximation," *Neural Networks*, vol. 45, pp. 134–143, 2013.

[199] M. Rehn and F. T. Sommer, "A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields," *Journal of Computational Neuroscience*, vol. 22, pp. 135–146, Oct 2007.

[200] L. Perrinet, M. Samuelides, and S. Thorpe, "Sparse spike coding in an asynchronous feed-forward multi-layer neural network using matching pursuit," *Neurocomputing*, vol. 57, pp. 125 – 134, 2004.

[201] J. Zylberberg, J. T. Murphy, and M. R. DeWeese, "A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of v1 simple cell receptive fields," *PLoS Comput Biol*, vol. 7, p. e1002250, 10 2011.

[202] T. Hu, A. Genkin, and D. B. Chklovskii, "A network of spiking neurons for computing sparse representations in an energy efficient way," *Neural computation*, vol. 24, no. 11, pp. 2852–2872, 2012.

[203] A. Balavoine, J. Romberg, and C. Rozell, "Convergence and rate analysis of neural networks for sparse approximation," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 9, pp. 1377–1389, 2012.

[204] L. V. S. Boyd, *Convex Optimization*. Cambridge University Press, 2004.

[205] C. Schlottmann and P. Hasler, "A highly dense, lowpower, programmable analog vector-matrix multiplier: The FPAA implementation," *IEEE Journal of Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 3, pp. 1–9, 2011.

[206] A. Borghi, J. Darbon, S. Peyronnet, T. F. Chan, and S. Osher., "A simple compressive sensing algorithm for parallel many-core architectures.," tech. rep., UCLA Computational and Applied Mathematics Technical Report, September 2008.

[207] M. Andrecut, "Fast GPU implementation of sparse signal recovery from random projections," *Engineering Letters*, vol. 17, no. 3, pp. 151–158, 2009.

[208] S. S. Vasanawala, M. T. Alley, B. A. Hargreaves, R. A. Barth, J. M. Pauly, and M. Lustig, "Improved pediatric MR imaging with compressive sensing," *Radiology*, vol. 256, pp. 607–616, Aug 2010.

[209] A. Tikhonov, "Regularization of incorrectly posed problems," *Soviet Math. Dokl*, vol. 4, no. 6, pp. 1624–1627, 1963.

[210] R. Saab, R. Chartrand, and O. Yilmaz, "Stable sparse approximations via nonconvex optimization," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal*, pp. 3885–3888, 2008.

[211] M. Elad, B. Matalon, and M. Zibulevsky, "Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization," *Applied and Computational Harmonic Analysis*, vol. 23, pp. 346–367, 2007.

[212] H. Zou, "The adaptive lasso and its oracle properties," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1418–1429, 2006.

[213] J. Fan, "Comments on "wavelets in statistics: A review" by a. antoniadis," *Statistical Methods and Applications*, vol. 6, pp. 131–138, Sep 1997.

[214] A. Antoniadis and J. Fan, "Regularization of wavelet approximations," *Journal od the American Statistical Association*, vol. 96, pp. 939–967, Sep 2001.

[215] M. Nikolova, "Local strong homogeneity of a regularized estimator," *SIAM Journal on Applied Mathematics*, vol. 61, no. 2, pp. 633–658, 2000.

[216] P. J. Huber, "Robust regression: Asymptotics, conjectures and Monte Carlo," *The Annals of Statistics*, vol. 1, pp. 799–821, Sep 1973.

[217] M. A. T. Figueiredo and R. D. Nowak, "Wavelet-based image estimation: An empirical Bayes approach using Jeffrey's noninformative prior," *IEEE Transactions on Image Processing*, vol. 10, pp. 1322–1331, Sep 2001.

[218] H. Gao, "Wavelet shrinkage denoising using the non-negative Garrote," *Journal of Computational and Graphical Statistics*, vol. 7, pp. 469–488, Dec 2001.

[219] C. F. Cadieu and B. A. Olshausen, "Learning intermediate-level representations of form and motion from natural movies," *Neural Computation*, vol. 24, no. 4, pp. 827–866, 2012.

[220] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relationships and efficient recovery," *IEEE Transactions on Signal Processing*, vol. 58, pp. 3042–3054, Mar 2010.

[221] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3869–3872, 2008.

[222] M. Khajehnejad, W. Xu, S. Avestimehr, and B. Hassibi, "Improved sparse recovery thresholds with two-step reweighted ℓ_1 minimization," *Arxiv preprint arXiv:1004.0402*, 2010.

[223] C. Twigg and P. Hasler, "Configurable analog signal processing," *Digital Signal Processing*, vol. 19, pp. 904–922, December 2009.

[224] A. S. Charles, A. Ahmed, A. Joshi, S. Conover, C. Turnes, and M. Davenport, "Cleaning up toxic waste: Removing nefarious contributions to recommendation systems," *Proceedings of ICASSP*, May 2013.

[225] R. Giryes, G. Sapiro, and A. M. Bronstein, "On the stability of deep networks," *arXiv preprint arXiv:1412.5896*, 2014.

[226] Y. Plan and R. Vershynin, "Dimension reduction by random hyperplane tessellations," *Discrete & Computational Geometry*, vol. 51, no. 2, pp. 438–461, 2014.

[227] B. Klartag and S. Mendelson, "Empirical processes and random projections," *Journal of Functional Analysis*, vol. 225, no. 1, pp. 229–245, 2005.

[228] W. Mantzel and J. Romberg, "Compressed subspace matching on the continuum," *arXiv preprint arXiv:1407.5234*, 2014.

[229] C. Ekanadham, D. Tranchina, and E. P. Simoncelli, "Recovery of sparse translation-invariant signals with continuous basis pursuit," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4735–4744, 2011.

[230] G. Tang, B. N. Bhaskar, P. Shah, and B. Recht, "Compressed sensing off the grid," *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7465–7490, 2013.

[231] K. C. Knudson, J. Yates, A. Huk, and J. W. Pillow, "Inferring sparse representations of continuous signals with continuous orthogonal matching pursuit," in *Advances in Neural Information Processing Systems*, pp. 1215–1223, 2014.

[232] D. Amelunxen, M. Lotz, M. B. McCoy, and J. A. Tropp, "Living on the edge: Phase transitions in convex programs with random data," *Information and Inference*, p. iau005, 2014.

[233] H. Rauhut and R. Ward, "Interpolation via weighted *l*_1 minimization," *arXiv preprint arXiv:1308.0759*, 2013.

[234] G. A. Miller, "The magical number seven, plus or minus two: some limits on our capacity for processing information.," *Psychological review*, vol. 63, no. 2, p. 81, 1956.

[235] S. J. Luck and E. K. Vogel, "The capacity of visual working memory for features and conjunctions," *Nature*, vol. 390, no. 6657, pp. 279–281, 1997.

[236] W. J. Ma, M. Husain, and P. M. Bays, "Changing concepts of working memory," *Nature neuroscience*, vol. 17, no. 3, pp. 347–356, 2014.

[237] T. Zhang, "Some sharp performance bounds for least squares regression with l1 regularization," *The Annals of Statistics*, vol. 37, no. 5A, pp. 2109–2144, 2009.

[238] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk, "Beyond Nyquist: efficient sampling of sparse bandlimited signals," *IEEE Trans. Inform. Theory*, vol. 56, Jan. 2009.

[239] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," in *Compressed Sensing, Theory and Applications* (Y. Eldar and G. Kutyniok, eds.), ch. 5, pp. 210–268, Cambridge Univ. Pr., Nov. 2012.

[240] V. H. De La Peña and E. Giné, *Decoupling: From Dependence to Independence*. Springer Verlag, 1999.

[241] M. Ledoux and M. Talagrand, *Probability in Banach Spaces: isoperimetry and processes*, vol. 23. Springer, 1991.

[242] D. Gross, "Recovering low-rank matrices from few coefficients in any basis," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1548–1566, 2011.

[243] J. A. Tropp, "User-friendly tail bounds for sums of random matrices," *Foundations of Computational Mathematics*, vol. 12, no. 4, pp. 389–434, 2012.