Title
Noisy Signal Correlation and Neural Network Pruning

Author
Moore, Eli

Publication Date
2023

Peer reviewed|Thesis/dissertation

Noisy Signal Correlation and Neural Network Pruning

By

ELI MOORE

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

APPLIED MATHEMATICS

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Professor Rishidev Chaudhuri, Chair

Professor Timothy Lewis

Professor Randall O'Reilly

Committee in Charge

2023

To my life partner, Gal.
Your infinite patience, steadfast loyalty, and undying love have fostered a bond stronger than I ever could have imagined. You deserve the world, and I'll do everything to ensure you have it.

To my mother, Pnina.
You sacrificed your life so that I could thrive, and never asked for anything in return but my success. You inspire me to never take my life for granted and to devote myself to those in need.

To my dearest friend, Ryan.
I never aspired to greatness until you showed me that it was possible, leading by example in every challenge you take on. You are a champion in every sense, truly rivaled by no one.

We are not here for ourselves, but for others.
You three are my others.

Contents

Abstract

Noise is ever-present in communication networks, both in technology and biology. Cell phone networks, radio systems, and the brain all exhibit signals accompanied by random fluctuations that seem to obfuscate transmitted data. Phone calls are dropped, radio signals compete with one another, and synaptic currents vary stochastically as neurons attempt to communicate. However, inherent randomness is not necessarily a nuisance — many randomized algorithms offer greater computational efficiency than their deterministic counterparts without sacrificing performance. What can be done to ensure that information is appropriately transmitted through noisy channels? In what contexts can noise be used as a tool for computation? These questions are explored throughout this dissertation.

In Chapter 2, I explore the problem of time-lagged sequence correlation minimization, which is critical for robust signal identification and synchronization in the presence of noise. I classify the family of sequence pairs that minimize a time-lagged correlation inequality and compute correlation properties of these sequences. In Chapter 3, I investigate synaptic pruning in the brain, a process in which unimportant neural connections are removed. I develop a noise-driven, biologically-plausible, unsupervised pruning algorithm with strong theoretical guarantees for a class of recurrent neural networks. In Chapter 4, I extend these ideas to pruning weights of nonlinear artificial neural networks used in machine learning. Here, I develop a randomized pruning algorithm that can be applied to a wider class of neural networks, and provide analytic error bounds for the output of feed-forward neural networks with pruned weight matrices.

Acknowledgments

Thank you to my advisor, Rishidev Chaudhuri. From the start of our work together, you offered encouragement, support, and a slew of ideas to help me develop as a scholar. Your patience and guidance, especially throughout the pandemic, helped me strengthen my resolve for mathematics and higher education as a whole. Thank you for years of jovial conversation and thinking about math together.

Thank you to my dissertation committee, Tim Lewis and Randall O'Reilly. In different ways, you both helped me carve my own niche as a mathematician.

Thank you to my teaching mentors, Jeffrey Anderson, Rohit Thomas, Korana Burke, and Ali Dad-Del. You have all been role models to me, and have undoubtedly had an effect on my teaching philosophy and practices.

Thank you to my students, past, present, and future. You are the reason I chose this profession. Helping you achieve your dreams is the unrivaled joy of my career.

Thank you to Tina Denena, for always being so helpful and willing to chat.

Thank you to my friends and fellow grads Jeffrey Nichols, Avishai Halev, Jeonghoon Kim, Raaghav Ramani, Kayden Mimmack, James Hughes, Vinh Nguyen, Ray Chou, Elysée Wilson-Egolf, Kyle Chickering, Yuan Ni, Shizhou Xu, Eric Severson, Sam Fleischer, Jenny Brown, Anandita De, Kayleigh Adams and many others. Our shared journey has provided much perspective and reassurance.

Thank you to my undergraduate professors Jing Li, Werner Horn, Daniel Katz, Katherine Stevenson, and Andrea Nemeth. I came to your department as an absolutely directionless and insecure, yet curious, student. You helped build my foundation and shaped me into the confident mathematician I am today. I hesitate to imagine where I would be without your collective efforts.

Thank you to my undergraduate peers Trevor Klar, David Angeles, Julian Corpeno, and so many others. I have an incredible number of fond memories doing and tutoring math together during all hours of the day. Thank you for being such a supportive and engaging community.

Now, thank you to my dearest friends and family. I am truly lucky to have such an enormous support network. This PhD is as much yours as it is mine.

Thank you to my mother, Pnina Moore, for being both my mom and my dad. Thank you for doing everything you could to raise me. Thank you for always encouraging me to succeed in school, even if it took until college for me to really listen. Thank you for supporting my dreams and my career choices. Thank you to Shabi Moore, for teaching me what not to do. Thank you to my brothers, Ron and Tal Moore, for never treating me any differently. Thank you to Greg Wolfus and donor family 1170, for getting to know me, and helping me get to know myself. Thank you to the Dimand family, for welcoming me with open arms.

Thank you to my childhood friends Ryan Machado, Max Downs, George Ashi, Giovanni Garcia, Sachin Sachdeva, and Alex Barr. You have cheered me on every step of the way. It has been such a pleasure supporting each other throughout our vastly different career paths.

This dissertation is written in loving memory of both my sister Zoe Wolfus and childhood friend Jessie Robles. Losing you both during the pandemic had a profound impact not only on my grad school experience, but my perspective surrounding life as a whole. Thank you for the time we got together.

Finally, thank you to my wife, Gal, for keeping me sane and providing undying love through the peaks and valleys of life. Thank you for your infinite patience. Thank you for believing in me. Thank you for being your wonderful self. I am so proud to be carving out a corner of the world together with you.

CHAPTER 1

Introduction

Both biological and digital communication networks are affected by noisy interference. Sequences with minimal correlation are used to circumvent noise in a multitude of applications such as communication networks, cryptography, acoustic design, and remote sensing [**28**, **43**, **46**]. For digital communication networks, finding signal encodings (sequences) that have minimal autocorrelation (a measure of correlation between a sequence and itself) aids in network synchronization, in the sense that a receiver can coherently recover the entirety of a sender's signal. Similarly, finding signal encodings with minimal crosscorrelation to other signals allows for more robust signal identification in the presence of noise, so that distinct senders are not confused for one another. Various studies have explored the simultaneous minimization of sequence autocorrelation and crosscorrelation [**41**, **44**, **45**, **77**]. When designing a communication system, one is free to choose the signal codes that represent users, so they should choose codes that are well-separated to reduce the impact of noisy interference. This code selection is the primary focus of Chapter 2, in which I classify the family of sequence pairs that minimize the so-called Pursley-Sarwate criterion [**68**], which is a combined measure of autocorrelation and crosscorrelation.

Unlike radio systems and cell phone networks, the brain is not a communication system designed by engineers; it is a biologically fine-tuned network that has evolved over millions of years in order to exhibit the behavior we are familiar with today. Notably, the brain maintains remarkable energy efficiency, incredibly sparse synaptic density, and a high degree of noise at multiple levels (e.g., via stochasticity in synaptic currents, large background fluctuations in neural activity, and varying responses to the same stimuli [**16**, **21**, **48**, **81**]). In fact, recent studies are suggesting that the energy efficiency of the brain and its noisy synaptic failures may be related [**78**]. Since we can not simply select the neural signals that the brain transmits in order to circumvent its inherent noise, we will instead study the relationship between these transmitted signals and their noisy fluctuations. We seek to further understand the role of noise in maintaining the brain's computational efficiency and

the extent to which it may contribute to synaptic density via pruning, which is the brain's tendency to remove unimportant connections. Noisy fluctuations will play a crucial role in developing the biologically-plausible algorithms described in Chapters 3 and 4.

The algorithms devised throughout Chapters 3 and 4 are inspired by the neuroscientific notion of synaptic pruning, which is the removal or weakening of unimportant synapses in the brain. Neuroscience and artificial intelligence initially developed as inextricably linked academic fields due to their joint origin stemming from the McCulloch-Pitts neuron, first described in 1943 [**58**], as well as decades of rapid collaborative productivity [**35**]. However, in recent years, these disciplines have increasingly found distinction as their disciplinary questions, tools, and practices continue to evolve. As such, the reciprocal relationship between them is no longer what it once was. Synaptic pruning has been explored in both biological and artificial settings, but with different goals and constraints. For example, artificial neural network pruning prioritizes faster computation and reduced storage requirements [**36**, **49**]. However, the state-of-the-art pruning algorithms tend not to be biologically plausible, and are thus unlikely to elucidate the brain's pruning mechanisms. If we study biologically-plausible pruning rules that facilitate efficient artificial neural network computation, we may be able to arrive at an explanation for the brain's sparse-yet-powerful structure. Conversely, if we can understand how the brain finds and maintains its sparse network structure, we can hope to replicate the remarkable energy efficiency it has in artificial neural networks.

Below is a brief overview of the chapters to follow.

1.1. Chapter Overview

Chapter 2. Pursley and Sarwate established a lower bound on a combined measure of autocorrelation and crosscorrelation for a pair of binary sequences [**68**]. Sequences that have low values of this combined correlation measure are useful for signal synchronization and identification, so they have been studied numerously [**41**, **44**, **45**, **77**]. Golay complementary pairs are a family of sequence pairs first discovered by Golay in 1949 for use in multislit spectrometry [**26**], and have since found application in physics, combinatorics, and telecommunications (see [**63**] for a recent review). I show that Pursley and Sarwate's lower bound is met precisely for Golay complementary pairs in the binary case. I also provide a generalization of their bound for when the sequence entries are

arbitrary complex numbers, notably extending their result to polyphase (n-ary) sequences. Finally, I investigate the autocorrelation and crosscorrelation distribution of various Golay complementary pair construction algorithms, and find that these distributions are asymptotically identical as the length of the sequences tends to infinity.

Chapter 3. Matrix sparsification is of interest to the neural network literature; when applied to the weight matrix of a neural network, it corresponds to synaptic weight pruning of the network. In 2011, Spielman and Srivastava developed a matrix sparsification algorithm that approximately preserves the spectrum of a limited class of matrices while discarding a majority of its entries [**83**]. While powerful, Spielman and Srivastava's algorithm requires computation of the (pseudo-)inverse of the input matrix, meaning that the importance of each synapse depends on global information. This is a property that biologically-plausible pruning algorithms can not afford, as rapid decay in neural activity patterns prevents meaningful long-range information flow between distant neurons. I generalize this pruning algorithm to a wider class of matrices while maintaining the theoretical sparsity and spectral guarantees. Next, I show that applying the generalized algorithm in the context of a linear (or rectified-linear) noise-driven recurrent neural network allows one to forego the matrix inverse computation, instead relying on an empirical covariance matrix. This converts the pruning rule to one that only requires information local to the synapse, thus indicating that the updated pruning rule is biologically-plausible. These results indicate that noise may play a fundamental probative role in synaptic pruning.

Chapter 4. Recurrent networks are natural candidates to consider pruning in Chapter 3 due to the restricted nature of the pruning algorithm described there. Namely, the theoretical guarantees in that chapter only apply to symmetric, thus square, matrices. Feed-forward networks are the dominant structures in machine learning and artificial intelligence, but they are comprised of a collection of dense rectangular weight matrices in general. Building upon an algorithm for square matrices devised by Drineas and Zousiaz in 2011 [**18**], I construct a pruning algorithm that applies to general rectangular matrices using the powerful matrix Bernstein inequality. This pruning rule also provides strong theoretical guarantees about the spectra of the pruned matrices while eliminating most of their entries. I then provide an analytic error bound comparing the output of a sparsified

feed-forward network to that of its original counterpart. This bound can be made arbitrarily small under mild conditions such as layer-wise sigmoid activation functions and unitary weight matrices. Lastly, I extend these error bounds to discrete-time recurrent networks with both vector and sequence-of-vector outputs.

CHAPTER 2

Sequence Pairs with Lowest Combined Autocorrelation and Crosscorrelation

Published in *IEEE Transactions on Information Theory* (July 2022).

Edited for this dissertation.

Joint work with:

Daniel J. Katz

Department of Mathematics

California State University, Northridge

Northridge, CA, 91330

daniel.katz@csun.edu

2.1. Abstract

Pursley and Sarwate established a lower bound on a combined measure of autocorrelation and crosscorrelation for a pair (f, g) of binary sequences (i.e., sequences with terms in $\{-1, 1\}$). If f is a nonzero sequence, then its autocorrelation demerit factor, $\mathrm{ADF}(f)$, is the sum of the squared magnitudes of the aperiodic autocorrelation values over all nonzero shifts for the sequence obtained by normalizing f to have unit Euclidean norm. If (f, g) is a pair of nonzero sequences, then their crosscorrelation demerit factor, $\mathrm{CDF}(f, g)$, is the sum of the squared magnitudes of the aperiodic crosscorrelation values over all shifts for the sequences obtained by normalizing both f and g to have unit Euclidean norm. Pursley and Sarwate showed that for binary sequences, the sum of $\mathrm{CDF}(f, g)$ and the geometric mean of $\mathrm{ADF}(f)$ and $\mathrm{ADF}(g)$ must be at least 1. For randomly selected pairs of long binary sequences, this quantity is typically around 2. In this paper, we show that Pursley and Sarwate's bound is met for binary sequences precisely when (f, g) is a Golay complementary pair. We also prove a generalization of this result for sequences whose terms are arbitrary complex

numbers. We investigate constructions that produce infinite families of Golay complementary pairs, and compute the asymptotic values of autocorrelation and crosscorrelation demerit factors for such families.

2.2. Introduction

In this paper, we are interested in aperiodic correlation of sequences, so we define a *sequence* to be a doubly-infinite list of complex numbers, $f = (f_j)_{j\in\mathbb{Z}} = (\ldots, f_{-1}, f_0, f_1, f_2, \ldots)$, in which only finitely many of the terms f_j are nonzero. The set of all sequences is a $\mathbb{C}$-vector space with the usual component-wise addition and $\mathbb{C}$-scalar multiplication. The *support* of the sequence f, written $\operatorname{supp}(f)$, is the set of $j \in \mathbb{Z}$ such that $f_j \neq 0$. We say that a subset S of $\mathbb{Z}$ is *contiguous* to mean that, whenever S contains integers a and b, it also contains every integer c that lies between a and b; the empty set is (vacuously) contiguous. We say that a sequence f is *contiguous* to mean that its support is contiguous. If $f = (f_j)_{j\in\mathbb{Z}}$ is a sequence, then the *length* of f, written $\operatorname{len} f$, is the size of the smallest contiguous set that includes $\operatorname{supp}(f)$; so the length of the zero sequence is zero, and otherwise $\operatorname{len} f = \max\operatorname{supp}(f) - \min\operatorname{supp}(f) + 1$. Thus, $\operatorname{len} f = |\operatorname{supp}(f)|$ if f is contiguous. Certain types of sequences are especially interesting for applications. A *unimodular sequence* f is a contiguous sequence where f_j is unimodular (i.e., $|f_j| = 1$) for every $j \in \operatorname{supp}(f)$. A unimodular sequence in which every nonzero term f_j is an mth root of unity is called an *m-ary sequence*, and a *binary sequence* is a 2-ary sequence, so that every nonzero term is in $\{-1, 1\}$. Although our main interest is in unimodular sequences, many of the constructions that are used to obtain unimodular sequences with good properties have non-unimodular sequences in intermediate steps, and this is why we must make these careful technical definitions and prove results that can handle sequences in general.

If $f = (f_j)_{j\in\mathbb{Z}}$ and $g = (g_j)_{j\in\mathbb{Z}}$ are sequences, then for $s \in \mathbb{Z}$, we define the *aperiodic crosscorrelation of f with g at shift s* to be

$$C_{f,g}(s) = \sum_{j\in\mathbb{Z}} f_{j+s}\overline{g_j}. \tag{2.1}$$

Only finitely many terms of the above sum can be nonzero due to the finite support of our sequences. The *aperiodic autocorrelation of f at shift s* is $C_{f,f}(s)$. For the rest of this paper, we simply say

crosscorrelation (resp., *autocorrelation*) to mean the aperiodic crosscorrelation (resp., aperiodic autocorrelation).

We write $\|f\|_2$ for the Euclidean norm of f, i.e., $\|f\|_2 = \sqrt{\sum_{j\in\mathbb{Z}} |f_j|^2}$. Then note that the autocorrelation at shift 0 is is the squared Euclidean norm of f; that is,

$$C_{f,f}(0) = \|f\|_2^2 ,$$

which is always a nonnegative real number, and is equal to $\operatorname{len} f$ if f is unimodular. If $f \neq 0$ and one wants a scaled version of f that is a unit vector with respect to the Euclidean norm, then one should divide the terms of f by $\|f\|_2 = \sqrt{C_{f,f}(0)}$ to obtain the *normalization of* f. We say that a pair (f, g) of sequences is *isoenergetic* to mean that f and g have the same Euclidean norm. Many applications use isoenergetic pairs, which include pairs consisting of unimodular sequences of the same length.

In applications, one is interested in pairs (f, g) of sequences where the crosscorrelation values of f with g at all shifts are small in magnitude, so that f and g are easily distinguished. Furthermore, one wants the autocorrelation values of f at all nonzero shifts to be small in magnitude, and similarly with g; this aids in synchronization. There are two main ways that these goals of achieving smallness of correlation are measured: l^∞ and l^2 methods. The l^∞ (worst-case) measures look at the largest magnitude among the undesirable correlations (i.e., autocorrelations of both sequences at nonzero shifts and all crosscorrelations between the two sequences). The l^∞ measure for autocorrelation of a sequence f is called the *peak sidelobe level (PSL)*, and gives the maximum of $|C_{f,f}(s)|$ over all nonzero $s \in \mathbb{Z}$. The l^∞ measure for crosscorrelation for a pair (f, g) is called the *peak crosscorrelation (PCC)*, and gives the maximum of $|C_{f,g}(s)|$ over all $s \in Z$. The l^2 measures of smallness of correlation can be considered mean square measures and are often called demerit factors (or, in reciprocal form, merit factors), and we shall define them in the next two paragraphs. Pursley **[67]**, Burr **[13]**, and Kärkkäinen **[41]** all express the view that the l^2 measure is a better indicator of performance than the l^∞ measure when evaluating the crosscorrelation of sequences in Code-Division Multiple Access (CDMA) applications.

For a pair (f, g) of nonzero sequences, the *crosscorrelation demerit factor of f with g* is defined by

$$\mathrm{CDF}(f, g) = \frac{\sum_{s\in\mathbb{Z}} |C_{f,g}(s)|^2}{C_{f,f}(0)C_{g,g}(0)}. \tag{2.2}$$

This is the sum of squared magnitudes of all the crosscorrelation values for the normalization of f with the normalization of g. Only finitely many terms of the sum are nonzero since $C_{f,g}(s) = 0$ whenever the shift s is not a difference between an element of $\mathrm{supp}(f)$ and an element of $\mathrm{supp}(g)$. Note that $\mathrm{CDF}(g, f) = \mathrm{CDF}(f, g)$ because $C_{g,f}(s) = \overline{C_{f,g}(-s)}$ for every $s \in \mathbb{Z}$. The *crosscorrelation merit factor of f with g* is the reciprocal of their crosscorrelation demerit factor.

For a nonzero sequence f, the *autocorrelation demerit factor of f* is defined by

$$\mathrm{ADF}(f) = \frac{\sum_{s\in\mathbb{Z}\smallsetminus\{0\}} |C_{f,f}(s)|^2}{C_{f,f}(0)^2} = -1 + \mathrm{CDF}(f, f). \tag{2.3}$$

This is the sum of squared magnitudes of all the autocorrelation values at nonzero shifts for the normalization of f. The *autocorrelation merit factor of f* is the reciprocal of its autocorrelation demerit factor.

Since we want pairs (f, g) of sequences with small magnitude autocorrelation values at nonzero shifts and small magnitude crosscorrelation values at all shifts, we want $\mathrm{ADF}(f)$, $\mathrm{ADF}(g)$, and $\mathrm{CDF}(f, g)$ all to be as small as possible. Pursley and Sarwate [**68**, eqs. (3),(4)] proved bounds involving these three quantities when f and g are nonzero binary sequences of the same length; their bounds are

$$-\sqrt{\mathrm{ADF}(f)\,\mathrm{ADF}(g)} \le \mathrm{CDF}(f, g) - 1 \le \sqrt{\mathrm{ADF}(f)\,\mathrm{ADF}(g)}. \tag{2.4}$$

In particular, the lower bound shows that

$$\sqrt{\mathrm{ADF}(f)\,\mathrm{ADF}(g)} + \mathrm{CDF}(f, g) \ge 1.$$

This places a limitation on how low we can simultaneously make all three demerit factors. Sarwate and Pursley later generalized their result to pairs of sequences with real terms [**77**, eqs. (8),(9)]. In Theorem 2.1 below, we show that Pursley and Sarwate's bounds hold for sequences whose terms are

arbitrary complex numbers, and the sequences need not be of the same length. This generalized bound could be obtained by going through Pursley and Sarwate's proof and making appropriate modifications, but we use a more efficient technique (based on Laurent polynomial representations of sequences) to obtain the key relations in full generality.

In view of this bound, we define the *Pursley–Sarwate criterion* of any pair (f, g) of nonzero sequences to be

$$\mathrm{PSC}(f, g) = \sqrt{\mathrm{ADF}(f)\,\mathrm{ADF}(g)} + \mathrm{CDF}(f, g),$$

so that

$$\mathrm{PSC}(f, g) \geq 1$$

for all pairs (f, g) of nonzero sequences.

Sarwate [**76**, eqs. (13),(38)] showed that a random binary sequence f of length ℓ (selected with uniform distribution) has an expected value for $\mathrm{ADF}(f)$ of $1 - 1/\ell$ and a randomly selected pair (f, g) of such sequences has an expected value for $\mathrm{CDF}(f, g)$ of 1. So for pairs of randomly selected long binary sequences, we expect $\mathrm{PSC}(f, g)$ to be around 2.

Since we want both autocorrelation and crosscorrelation to be as low as possible, we would like to know which pairs (f, g) of sequences have $\mathrm{PSC}(f, g) = 1$. In fact, we find a complete classification of such pairs. A pair of sequences (f, g) with $C_{f,f}(s) + C_{g,g}(s) = 0$ for all nonzero $s \in \mathbb{Z}$ is called a *Golay complementary pair* in honor of Golay who introduced them in [**26**]. It turns out that Golay complementarity is the key to achieving a Pursley–Sarwate criterion equal to 1.

Theorem 2.1. *Let f and g be nonzero sequences. Then*

$$-\sqrt{\mathrm{ADF}(f)\,\mathrm{ADF}(g)} \leq \mathrm{CDF}(f, g) - 1 \leq \sqrt{\mathrm{ADF}(f)\,\mathrm{ADF}(g)}.$$

Both of these inequalities simultaneously become equalities if and only if we have $\min\{\operatorname{len} f, \operatorname{len} g\} = 1$, in which case $\mathrm{PSC}(f, g) = 1$. If $\min\{\operatorname{len} f, \operatorname{len} g\} > 1$, then equality is achieved in the lower bound (i.e., $\mathrm{PSC}(f, g) = 1$) if and only if there is some $\lambda \in \mathbb{C}$ such that $(f, \lambda g)$ is a Golay complementary pair, in which case $\lambda \neq 0$ and $(f, |\lambda| g)$ is also a Golay complementary pair. In particular, if $\min\{\operatorname{len} f, \operatorname{len} g\} > 1$ and f and g are unimodular, then equality is achieved in the lower bound if and only if (f, g) is a Golay pair, in which case $\operatorname{len} f = \operatorname{len} g$ and $\mathrm{ADF}(f) = \mathrm{ADF}(g)$.

It should be noted that Pursley and Sarwate came close to this result, which comes from a Cauchy–Schwarz inequality. In [**68**], they state that the Cauchy–Schwarz inequality becomes an equality if and only if the two vectors are equal, and later in [**77**] they state that it becomes an equality if and only if the first vector is a scalar multiple of the second. (The correct necessary and sufficient condition is that one of the vectors must be a scalar multiple of the other, so Pursley and Sarwate's second formulation is much closer to being correct than their first, but it still neglects the possibility that equality can occur when the second vector is zero and the first vector is nonzero.) Pursley and Sarwate's first formulation of the necessary and sufficient condition for equality in the Cauchy–Schwarz inequality led them to state that it is impossible [**68**, p. 305] to meet the lower bound in (2.4). When they realized that this first formulation was incorrect and stated the second formulation, they noted [**77**, p. 49] the unsoundness of their earlier argument that asserted the impossibility of meeting the lower bound in (2.4). But their second paper still does not show that the lower bound is actually met, although they deduce the correct condition that would need to be met in the case of binary sequences.

Once Theorem 2.1 is established, Turyn's construction [**92**, Corollary to Lemma 5] supplies Golay pairs for infinitely many lengths, so we obtain infinitely many pairs (f, g) of binary sequences with $\mathrm{PSC}(f, g) = 1$.

Corollary 2.2. *If $\ell = 2^a 10^b 26^c$ for some nonnegative integers a, b, c, then there is a pair (f, g) of binary sequences, with each sequence of length ℓ, such that* $\mathrm{PSC}(f, g) = 1$.

Theorem 2.1 tells us that the pairs (f, g) of unimodular sequences of length greater than 1 that have $\mathrm{PSC}(f, g) = 1$ are precisely Golay complementary pairs. We are interested in the relative magnitudes of $\mathrm{ADF}(f)$, $\mathrm{ADF}(g)$, and $\mathrm{CDF}(f, g)$ in Golay pairs.

Golay pairs are usually constructed by means of various transformations and combination rules starting from a small set of initial pairs. We mention some of the simplest construction methods here. If we have a sequence $f = (f_j)_{j\in\mathbb{Z}}$ with $\mathrm{supp}(f) \subseteq \{0, 1, \ldots, \mathrm{len}\, f - 1\}$, then the *conjugate reverse of* f, written $f^\ddagger$, is the sequence with $(f^\ddagger)_j = \overline{f_{\mathrm{len}\, f-1-j}}$ for $j \in \{0, 1, \ldots, \mathrm{len}\, f - 1\}$ and $f^\ddagger_j = 0$ for all other j. If both $f = (f_j)_{j\in\mathbb{Z}}$ and $g = (g_j)_{j\in\mathbb{Z}}$ are sequences of length ℓ whose supports are subsets of $\{0, 1 \ldots, \ell - 1\}$, then the *concatenation of f with g*, written $f|g$, is the sequence $c = (c_j)_{j\in\mathbb{Z}}$ of length

2ℓ with $c_j = f_j$ for $j \in \{0, 1, \ldots, \ell - 1\}$, with $c_j = g_{j-\ell}$ for $j \in \{\ell, \ell + 1, \ldots, 2\ell - 1\}$, and with $c_j = 0$ for all other j. The *interleaving of f with g*, written $f \wr g$, is the sequence $h = (h_j)_{j \in \mathbb{Z}}$ of length 2ℓ with $h_{2j} = f_j$ and $h_{2j+1} = g_j$ for $j \in \{0, 1, \ldots, \ell - 1\}$, and with $h_k = 0$ for $k \notin \{0, 1 \ldots, 2\ell - 1\}$.

The *Golay–Rudin–Shapiro recursion*, as typically employed, begins with an isoenergetic Golay pair $(f^{(0)}, g^{(0)})$ (known as a *seed pair*), both of whose sequences are of the same positive length, ℓ, and have supports that are subsets of $\{0, 1, \ldots, \ell - 1\}$. Then the Golay–Rudin–Shapiro recursion produces a family $(f^{(n)}, g^{(n)})_{n \geq 0}$ of Golay pairs by the rule $f^{(n+1)} = f^{(n)} | g^{(n)}$ and $g^{(n+1)} = f^{(n)} | - g^{(n)}$. Thus, the length of sequences doubles at each step, so that $f^{(n)}$ and $g^{(n)}$ are sequences of length $2^n \ell$. If the seed pair has sequences that are unimodular (resp., binary), then the entire family will consist of unimodular (resp., binary) sequences. If one begins with the seed pair $(f^{(0)}, g^{(0)})$ where both sequences are of length 1 with their nonzero terms equal to 1, then the construction produces the Rudin–Shapiro sequences. One consequence of our Theorem 2.40 is that we know the asymptotic behavior of the demerit factors for such a family.

Theorem 2.3. *Let $(f^{(n)}, g^{(n)})_{n \geq 0}$ be a family of Golay pairs produced by the Golay–Rudin–Shapiro recursion as described above. Then*

$$\lim_{n \to \infty} \mathrm{ADF}(f^{(n)}) = \lim_{n \to \infty} \mathrm{ADF}(g^{(n)}) = 1/3, \text{ and}$$
$$\lim_{n \to \infty} \mathrm{CDF}(f^{(n)}, g^{(n)}) = 2/3.$$

The *simple Golay interleaving recursion*, as typically employed, also begins with a seed Golay pair $(f^{(0)}, g^{(0)})$ that is isoenergetic, with both sequences of the same positive length, ℓ, and having supports that are subsets of $\{0, 1, \ldots, \ell - 1\}$. The simple Golay interleaving recursion produces a family $(f^{(n)}, g^{(n)})_{n \geq 0}$ of Golay pairs by the rule $f^{(n+1)} = f^{(n)} \wr g^{(n)}$ and $g^{(n+1)} = g^{(n)\ddagger} \wr -f^{(n)\ddagger}$. Thus the length of sequences doubles at each step, so that $f^{(n)}$ and $g^{(n)}$ are sequences of length $2^n \ell$. If the seed pair has sequences that are unimodular (resp., binary), then the entire family will consist of unimodular (resp., binary) sequences. Our Theorem 2.54 shows that such families have the same asymptotic behavior as those produced by the Golay–Rudin–Shapiro recursion.

Theorem 2.4. *Let $(f^{(n)}, g^{(n)})_{n\geq 0}$ be a family of Golay pairs produced by the simple Golay interleaving recursion as described above. Then*

$$\lim_{n\to\infty} \mathrm{ADF}(f^{(n)}) = \lim_{n\to\infty} \mathrm{ADF}(g^{(n)}) = 1/3, \text{ and}$$
$$\lim_{n\to\infty} \mathrm{CDF}(f^{(n)}, g^{(n)}) = 2/3.$$

In fact, our Theorems 2.40 and 2.54 show that we obtain the same asymptotic behavior described in Theorems 2.3 and 2.4 here even if we allow ourselves much greater freedom in constructing the infinite families of pairs than indicated here. This freedom comes by applying a transformation to each pair $(f^{(n)}, g^{(n)})$ before applying the recursion rule that doubles the length of the sequences. These transformations are inspired by the work of Golay, who showed that there are 64 transformations of a binary pair (f, g) that preserve the Golay complementary property: these include exchanging f for g, negating f, negating g, reversing f, reversing g, negating the terms with odd indices in both f and g, and compositions of any selection of these transformations. These transformations also work for Golay pairs with complex-valued sequences, provided that one uses the conjugate reverse operation as the generalization of Golay's reverse operation. Theorem 2.40 shows that one still gets the same limiting values for demerit factors as attested in Theorem 2.3 if one applies such transformations at each stage of the Golay–Rudin–Shapiro recursion before the next doubling of length by concatenation (and one may use different transformations at different stages). Theorem 2.54 shows that the interleaving construction also exhibits the same limiting behavior as in Theorem 2.4 when one uses the transformations between doublings, although it places the restriction that whenever one conjugate reverses the first sequence in a pair, one must also conjugate reverse the second sequence. Even now, we have not indicated the full scope of Theorems 2.40 and 2.54, which allow for more general choices of seed Golay pairs and transformations to be used at each step. It should also be noted that Theorems 2.40 and 2.54 provide exact formulae for $\mathrm{ADF}(f^{(n)})$, $\mathrm{ADF}(g^{(n)})$, and $\mathrm{CDF}(f^{(n)}, g^{(n)})$ for each n, from which one obtains the asymptotic results given in Theorems 2.3 and 2.4 here.

This paper is organized as follows. Section 2.3 introduces the formalism of viewing sequences as Laurent polynomials, and provides the notations and conventions for the rest of the paper, along with proofs of some preliminary facts. Section 2.4 is the proof of Theorem 2.1. Section 2.5 describes a group that generalizes Golay's collection of 64 transformations that preserve complementarity

of pairs; this group is very useful in our proofs on the asymptotic behavior of demerit factors. Section 2.6 describes a single construction for complex Golay pairs that is more general and simpler to use than previous constructions of Golay and Turyn; this eases our proofs of asymptotic results. 2.7 has the proof 2.40 (of which 2.3 here is one corollary) on the behavior of demerit factors for families produced by the Golay–Rudin–Shapiro recursion. Section 2.8 has the proof Theorem 2.54 (of which Theorem 2.4 here is one corollary) on the behavior of demerit factors for families produced by the simple Golay interleaving recursion. Section 2.9 poses an open problem that asks whether all families of Golay pairs consisting of binary sequences whose lengths tend to infinity have asymptotic autocorrelation and crosscorrelation demerit factors that tend to 1/3 and 2/3, respectively. The same question is also asked of the more general class of Golay pairs consisting of unimodular sequences.

2.3. Preliminaries

Recall from the Introduction the definition of a sequence, its support, its length, its Euclidean norm, and its normalization. Also recall what it means for a subset of $\mathbb{Z}$ or a sequence to be contiguous, as well as the definitions of contiguous, unimodular, m-ary, and binary sequences, and of isoenergetic sequence pairs. We also continue to use the definitions of crosscorrelation and autocorrelation, with their respective demerit factors, and the Pursley–Sarwate criterion. One should also recall the definition of Golay complementary pairs, which are also simply called *Golay pairs* or *complementary pairs*. All these definitions and their notations from the Introduction remain in force throughout this paper. We use $\mathbb{N}$ to denote the set $\{0, 1, \ldots\}$ of nonnegative integers.

We identify the sequence $f = (f_j)_{j\in\mathbb{Z}} = (\ldots, f_{-1}, f_0, f_1, f_2, \ldots)$ with the Laurent polynomial $f(z) = \sum_{j\in\mathbb{Z}} f_j z^j$ in $\mathbb{C}[z, z^{-1}]$, so that the definitions and notations for support, contiguity, length, unimodularity, m-arity, binarity, Euclidean norm, being isoenergetic, correlation, demerit factors, the Pursley–Sarwate criterion, and Golay complementarity all apply equally well to Laurent polynomials. We also use the convention that any property defined for a sequence (Laurent polynomial) can be predicated of a pair, in which case this is understood to mean that both sequences in the pair have that property. So, for example, a unimodular Golay pair (f, g) with $\operatorname{len}(f, g) = 10$ is a Golay pair (f, g) where f and g are both unimodular and $\operatorname{len} f = \operatorname{len} g = 10$. A *monomial* is sequence of length 1, that is, some cz^j where c is a nonzero complex number and $j \in \mathbb{Z}$.

If $f(z) \in \mathbb{C}[z, z^{-1}]$ is a nonzero Laurent polynomial, then its *order*, written $\operatorname{ord} f$, is the smallest j such that $f_j \neq 0$, while its *degree*, written $\deg f$, is the largest k such that $f_k \neq 0$. We create two new symbols, ∞ and $-\infty$, and decree that $\operatorname{ord} 0 = \infty$ and $\deg 0 = -\infty$. We extend the addition operation from $\mathbb{Z}$ to $\mathbb{Z} \cup \{\infty, -\infty\}$ by the following rules: (I) $a + \infty = \infty + a = \infty$ for $a \in \mathbb{Z} \cup \{\infty\}$, (II) $b + (-\infty) = (-\infty) + b = -\infty$ for $b \in \mathbb{Z} \cup \{-\infty\}$, and (III) $(-\infty) + \infty = \infty + (-\infty) = 0$. For every $f, g \in \mathbb{C}[z, z^{-1}]$, the first rules makes $\operatorname{ord}(fg) = \operatorname{ord} f + \operatorname{ord} g$, the second rule makes $\deg(fg) = \deg f + \deg g$, and the third rule makes $\operatorname{ord} f(z) + \deg f(z^{-1}) = 0$. Note that $\operatorname{len} f = \deg f - \operatorname{ord} f + 1$ for every nonzero f in $\mathbb{C}[z, z^{-1}]$.

If $f(z) = \sum_{j \in \mathbb{Z}} f_j z^j \in \mathbb{C}[z, z^{-1}]$, then we write $\overline{f(z)}$ to mean $\sum_{j \in \mathbb{Z}} \overline{f_j} z^{-j}$, where the inversion of z comes about because we are interested in our polynomials on the complex unit circle. We sometimes use f as a shorthand for $f(z)$, and in this case $\overline{f}$ is a shorthand for $\overline{f(z)}$. We do not need or introduce any notation for the operation that simply conjugates the coefficients of a Laurent polynomial without also inverting the indeterminate. We also use the convention that if $f(z) \in \mathbb{C}[z, z^{-1}]$, then $|f(z)|^2 = f(z)\overline{f(z)}$, and indeed, if k is any nonnegative integer, then $|f(z)|^{2k} = (f(z)\overline{f(z)})^k$. When we abbreviate $f(z)$ by f, then $|f|^2$ and $|f|^{2k}$ stand for $|f(z)|^2$ and $|f(z)|^{2k}$, respectively. Along the same lines, if $f(z) \in \mathbb{C}[z, z^{-1}]$, then $\operatorname{Re} f(z)$ is a shorthand for $(f(z) + \overline{f(z)})/2$ (which can be abbreviated $\operatorname{Re} f = (f + \overline{f})/2$).

For any $f(z) \in \mathbb{C}[z, z^{-1}]$, we use the convention that f_s is the coefficient of z^s in $f(z)$. Sometimes we use enclosing parentheses when the Laurent polynomial has a complicated form, for example if $f(z)$ and $g(z)$ are Laurent polynomials, then $(fg)_s$ is the coefficient of z^s in the product $f(z)g(z)$. If we write f_s^k, then we mean the coefficient of z^s in $f(z)^k$, that is, we mean $(f^k)_s$. And similarly $|f|_s^{2k}$ means the coefficient of z^s in $|f(z)|^{2k}$, that is $(|f|^{2k})_s$. Thus if $c \in \mathbb{C}$ and n is a nonzero integer, then $f(cz^n)_{nm} = c^m f_m$ for every $m \in \mathbb{Z}$; in particular $f(cz^n)_0 = f_0$. If, in addition, c is unimodular, then $|f(cz^n)|^2_{mn} = c^m |f|^2_m$ for every $m \in \mathbb{Z}$; in particular $|f(cz^n)|^2_0 = |f|^2_0$.

We now show how the correlation concepts, defined for sequences in the Introduction, are realized in the Laurent polynomial interpretation. If $f(z), g(z) \in \mathbb{C}[z, z^{-1}]$, we define the *crosscorrelation of f with g at shift s* to be

$$C_{f,g}(s) = (f\overline{g})_s, \tag{2.5}$$

which agrees with (2.1) when f and g are Laurent polynomials representing sequences. So the Laurent polynomial $f\overline{g}$ records all the crosscorrelation values:

$$f\overline{g} = \sum_{s\in\mathbb{Z}} C_{f,g}(s) z^s. \tag{2.6}$$

When $f = g$, we call $C_{f,f}(s)$ the *autocorrelation of f at shift s*. Note that

$$C_{f,f}(0) = |f|_0^2 = \sum_{s\in\mathbb{Z}} |f_s|^2 = \|f\|_2^2\,, \tag{2.7}$$

which is the squared Euclidean norm of f. Thus, a pair $(f, g) \in \mathbb{C}[z, z^{-1}]^2$ is isoenergetic if and only if $C_{f,f}(0) = C_{g,g}(0)$. We record some basic facts about correlation.

Lemma 2.5. *Let $f(z), g(z) \in \mathbb{C}[z, z^{-1}]$ and $s \in \mathbb{Z}$.*

(i). $C_{f,f}(0) = |f|_0^2 = \|f\|_2^2 = 0$ if and only if $f = 0$; otherwise $C_{f,f}(0)$ is positive real.

(ii). If f is unimodular, then $C_{f,f}(0) = |f|_0^2 = \|f\|_2^2 = \operatorname{len} f$.

(iii). $C_{g,f}(s) = \overline{C_{f,g}(-s)}$.

(iv). $C_{f,f}(s) = \overline{C_{f,f}(-s)}$.

(v). If $f, g \neq 0$ and either $s > \deg f - \operatorname{ord} g$ or $s < \operatorname{ord} f - \deg g$, then $C_{f,g}(s) = 0$.

(vi). If $f, g \neq 0$, then $C_{f,g}(\deg f - \operatorname{ord} g) = f_{\deg f}\overline{g_{\operatorname{ord} g}} \neq 0$ and $C_{f,g}(\operatorname{ord} f - \deg g) = f_{\operatorname{ord} f}\overline{g_{\deg g}} \neq 0$.

(vii). If $|s| \geq \operatorname{len} f$, then $C_{f,f}(s) = 0$.

(viii). If $f \neq 0$, then $C_{f,f}(\operatorname{len} f - 1) = f_{\deg f}\overline{f_{\operatorname{ord} f}} \neq 0$ and $C_{f,f}(1 - \operatorname{len} f) = f_{\operatorname{ord} f}\overline{f_{\deg f}} \neq 0$.

Proof. One immediately obtains part (i) from (2.7), and part (ii) also follows from (2.7) since the sum of the squared magnitudes of $\operatorname{len} f$ unimodular numbers is $\operatorname{len} f$. From (2.5), we see that $\overline{C_{f,g}(-s)} = \overline{(f\overline{g})_{-s}} = (\overline{f\overline{g}})_s = (g\overline{f})_s$, which by another application of (2.5) is $C_{g,f}(s)$. This proves part (iii), and if one sets $g = f$ there, then part (iv) follows. If $f, g \neq 0$, then the highest degree terms in f and $\overline{g}$ are $f_{\deg f} z^{\deg f}$ and $\overline{g_{\operatorname{ord} g}} z^{-\operatorname{ord} g}$, respectively, while the lowest degree terms in f and $\overline{g}$ are $f_{\operatorname{ord} f} z^{\operatorname{ord} f}$ and $\overline{g_{\deg g}} z^{-\deg g}$, respectively. Thus (2.5) shows that $C_{f,g}(s) = 0$ if $s > \deg f - \operatorname{ord} g$ or $s < \operatorname{ord} f - \deg g$, and shows that $C_{f,g}(\deg f - \operatorname{ord} g)$ and $C_{f,g}(\operatorname{ord} f - \deg g)$ have the desired values. This proves part (v) and part (vi), and if one sets $g = f$ in these and recalls that $\operatorname{len} f = \deg f - \operatorname{ord} f + 1$ and notes that $C_{0,0}(s) = 0$ for all $s \in \mathbb{Z}$, then one obtains part (vii) and part (viii). □

We can use (2.6) and (2.7) together to obtain

$$|fg|_0^2 = \sum_{s\in\mathbb{Z}} |C_{f,g}(s)|^2. \tag{2.8}$$

Then comparing expressions (2.8) and (2.7) with the terms in the definition (2.2) of the crosscorrelation demerit factor, we see that if $f, g \neq 0$, then

$$\mathrm{CDF}(f,g) = \frac{|fg|_0^2}{|f|_0^2|g|_0^2}, \tag{2.9}$$

and thus by (2.3)

$$\mathrm{ADF}(f) = -1 + \frac{|f|_0^4}{(|f|_0^2)^2}. \tag{2.10}$$

Expressions (2.9) and (2.10) connect crosscorrelation and autocorrelation demerit factors to L^p norms of polynomials on the complex unit circle; see [**44**, p. 515–516] for more details. We record a simple criterion for vanishing autocorrelation demerit factor.

Lemma 2.6. *Let f be a nonzero sequence. Then* $\mathrm{ADF}(f) = 0$ *if and only if* $\operatorname{len} f = 1$.

Proof. Parts (vii) and (viii) of Lemma 2.5 show that $C_{f,f}(s) = 0$ for all nonzero s if and only if $\operatorname{len} f \leq 1$. Since f is nonzero, this means that the numerator of $\mathrm{ADF}(f)$ in the expression in (2.3) is zero if and only if $\operatorname{len} f = 1$. □

The following lemma translates the definition of Golay complementary pair to the Laurent polynomial interpretation.

Lemma 2.7. *If $(f,g) \in \mathbb{C}[z,z^{-1}]^2$, then (f,g) is a Golay pair if and only if $|f|^2 + |g|^2$ is constant, in which case $|f|^2 + |g|^2 = |f|_0^2 + |g|_0^2$.*

Proof. By (2.6), we have $|f|^2 + |g|^2 = \sum_{s\in\mathbb{Z}} (C_{f,f}(s) + C_{g,g}(s))z^s$, which is constant if and only if $C_{f,f}(s) + C_{g,g}(s) = 0$ for all nonzero s, that is, if and only if (f,g) is a Golay pair. And $|f|^2 + |g|^2$ is constant if and only if it equals its own constant term, which is $|f|_0^2 + |g|_0^2$. □

We should note that, with trivial exceptions, sequences that form a Golay pair must be the same length.

Lemma 2.8. *If (f, g) is a Golay pair, then either* $\operatorname{len} f = \operatorname{len} g$ *or else* $\{\operatorname{len} f, \operatorname{len} g\} = \{0, 1\}$.

PROOF. Suppose that (f, g) is a Golay pair. If one sequence has a length ℓ with $\ell > 1$, then Lemma 2.5(viii) shows that its autocorrelation at shift $\ell - 1$ is nonzero, so that the Golay condition forces the other sequence to have nonzero autocorrelation at shift $\ell - 1$, which by Lemma 2.5(vii) forces it to have length at least ℓ. Thus, if either sequence has length greater than 1, they both must have the same length. □

We should also note that in many Golay pairs of interest, the autocorrelation demerit factors of the two elements are the same.

Lemma 2.9. *Let (f, g) be a Golay pair with $f, g \neq 0$. If (f, g) is isoenergetic, then* $\mathrm{ADF}(f) = \mathrm{ADF}(g)$. *If* $\mathrm{ADF}(f) = \mathrm{ADF}(g)$, *then (f, g) is isoenergetic or* $\operatorname{len} f = \operatorname{len} g = 1$.

PROOF. Since (f, g) is a Golay pair, we have $C_{f,f}(s) = -C_{g,g}(s)$ for every nonzero s, and so $\sum_{s \in \mathbb{Z} \smallsetminus \{0\}} |C_{f,f}(s)|^2 = \sum_{s \in \mathbb{Z} \smallsetminus \{0\}} |C_{g,g}(s)|^2$. Thus, if (f, g) is isoenergetic, then $C_{f,f}(0) = C_{g,g}(0)$, and then (2.3) shows that $\mathrm{ADF}(f) = \mathrm{ADF}(g)$. Conversely, if $\mathrm{ADF}(f) = \mathrm{ADF}(g)$, then (2.3) shows that either $C_{f,f}(0) = C_{g,g}(0)$ (so that (f, g) is isoenergetic) or $\mathrm{ADF}(f) = \mathrm{ADF}(g) = 0$, the latter of which can only hold if $\operatorname{len} f = \operatorname{len} g = 1$ by Lemma 2.6. □

Unimodular Golay pairs are of particular interest and have many of the good properties discussed above.

Lemma 2.10. *If f, g are nonzero unimodular sequences such that (f, g) is a Golay pair, then* $\operatorname{len} f = \operatorname{len} g = \|f\|_2^2 = \|g\|_2^2$ *(so that (f, g) is isoenergetic),* $|f|^2 + |g|^2 = 2 \operatorname{len} f = 2 \operatorname{len} g$, *and* $\mathrm{ADF}(f) = \mathrm{ADF}(g)$.

PROOF. Lemma 2.8 shows that $\operatorname{len} f = \operatorname{len} g$. Then Lemma 2.5(ii) shows that $\|f\|_2^2 = |f|_0^2 = \operatorname{len} f = \operatorname{len} g = |g|_0^2 = \|g\|_2^2$, so that (f, g) is isoenergetic, and then Lemma 2.7 shows that $|f|^2 + |g|^2 = 2 \operatorname{len} f$, while Lemma 2.9 shows that $\mathrm{ADF}(f) = \mathrm{ADF}(g)$. □

2.4. Proof of Theorem 2.1

We now prove our first main result, Theorem 2.1. If $\operatorname{len} f = 1$ (resp., $\operatorname{len} g = 1$), then Lemma 2.6 shows that $\mathrm{ADF}(f) = 0$ (resp., $\mathrm{ADF}(g) = 0$) and one easily calculates $\mathrm{CDF}(f, g) = 1$ from (2.9), so

that both inequalities are achieved simultaneously and we have $\mathrm{PSC}(f,g) = 1$. So henceforth we assume that $\min\{\operatorname{len} f, \operatorname{len} g\} > 1$.

Let $\Gamma = (f g \overline{f g})_0$, which we can interpret in two different ways. If we group the terms as $\Gamma = (|f|^2 \overline{|g|^2})_0$, and use (2.6) to interpret both $|f|^2 = f\overline{f}$ and $|g|^2 = g\overline{g}$ as sequences whose terms are autocorrelation values, then (2.5) shows that Γ is the crosscorrelation of the autocorrelation spectrum of f with the autocorrelation spectrum of g at shift 0, i.e., $\Gamma = \sum_{s\in\mathbb{Z}} C_{f,f}(s)\overline{C_{g,g}(s)}$. If instead we choose to group the terms as $\Gamma = |f\overline{g}|_0^2$, then (2.7) shows that Γ is the sum of the squared magnitudes of the coefficients of $f\overline{g}$, so by (2.6) we have $\Gamma = \sum_{s\in\mathbb{Z}} |C_{f,g}(s)|^2$, which shows that Γ is real. Now define $\Delta = -C_{f,f}(0)C_{g,g}(0) + \Gamma$. Since Γ, $C_{f,f}(0)$, and $C_{g,g}(0)$ are real, Δ is a real number with

$$\begin{aligned} \Delta &= -C_{f,f}(0)C_{g,g}(0) + \sum_{s\in\mathbb{Z}} |C_{f,g}(s)|^2 \\ &= \sum_{s\in\mathbb{Z}\smallsetminus\{0\}} C_{f,f}(s)\overline{C_{g,g}(s)}. \end{aligned} \tag{2.11}$$

Now the Cauchy–Schwarz inequality tells us that

$$|\Delta| \le \sqrt{\left(\sum_{s\in\mathbb{Z}\smallsetminus\{0\}} |C_{f,f}(s)|^2\right)\left(\sum_{s\in\mathbb{Z}\smallsetminus\{0\}} |C_{g,g}(s)|^2\right)}. \tag{2.12}$$

Use the first equality in (2.11) to substitute for Δ and divide by $C_{f,f}(0)C_{g,g}(0)$ (which is positive real since $f, g \neq 0$), to obtain the equivalent inequality

$$|-1 + \mathrm{CDF}(f,g)| \le \sqrt{\mathrm{ADF}(f)\,\mathrm{ADF}(g)}, \tag{2.13}$$

which is the bound we were to show.

The bound in the Cauchy–Schwarz inequality (2.12) and in the equivalent result (2.13) is met if and only if there is some nonzero $\mu \in \mathbb{C}$ such that $C_{f,f}(s) = \mu C_{g,g}(s)$ for every $s \in \mathbb{Z}\smallsetminus\{0\}$. (In principle, it would also be met if $C_{f,f}(s) = 0$ for all $s \in \mathbb{Z}\smallsetminus\{0\}$ or if $C_{g,g}(s) = 0$ for all $s \in \mathbb{Z}\smallsetminus\{0\}$, but these are impossible by Lemma 2.5(viii) because $\operatorname{len} f, \operatorname{len} g > 1$.) If we have such a μ, then we

have

$$\Delta = \sum_{s \in \mathbb{Z} \smallsetminus \{0\}} C_{f,f}(s)\overline{C_{g,g}(s)} = \mu \sum_{s \in \mathbb{Z} \smallsetminus \{0\}} |C_{g,g}(s)|^2, \tag{2.14}$$

and since Δ was shown to be real, this forces μ to be a real number.

If we have equality in (2.13) and μ is positive (resp., negative), then (2.14) shows that Δ is positive (resp., negative), so that $\Delta/(C_{f,f}(0)C_{g,g}(0)) = -1 + \mathrm{CDF}(f,g)$ is positive (resp., negative), and so $-1 + \mathrm{CDF}(f,g) = \sqrt{\mathrm{ADF}(f)\,\mathrm{ADF}(g)}$ and we meet the upper bound (resp., $-1 + \mathrm{CDF}(f,g) = -\sqrt{\mathrm{ADF}(f)\,\mathrm{ADF}(g)}$ and we meet the lower bound). These two extremes cannot be achieved simultaneously, as $\mathrm{ADF}(f)\,\mathrm{ADF}(g) \neq 0$ because $\operatorname{len} f > 1$ and $\operatorname{len} g > 1$ (see Lemma 2.6). So, when $\min\{\operatorname{len} f, \operatorname{len} g\} > 1$, the necessary and sufficient condition for achieving the lower bound $-1 + \mathrm{CDF}(f,g) = -\sqrt{\mathrm{ADF}(f)\,\mathrm{ADF}(g)}$ (i.e., $\mathrm{PSC}(f,g) = 1$) is for there to be a negative real number μ such that $C_{f,f}(s) = \mu C_{g,g}(s)$ for all nonzero $s \in \mathbb{Z}$, which is to say, such that $(f, \sqrt{-\mu}g)$ is a Golay pair. So to meet the lower bound, it is necessary that there be some $\lambda \in \mathbb{C}$ such that $(f, \lambda g)$ is a Golay pair. Conversely, if there is a $\lambda \in \mathbb{C}$ such that $(f, \lambda g)$ is a Golay pair, then we know that $\lambda \neq 0$ by Lemma 2.8, since $\operatorname{len} f > 1$. Since (2.5) shows that scalar multiplication of a sequence by a unimodular complex number (e.g., $|\lambda|/\lambda$) does not change its autocorrelation values, $(f, |\lambda|g) = (f, \sqrt{-(-|\lambda|^2)}g)$ is also a Golay pair with $-|\lambda|^2$ a negative real number, so we meet the lower bound by what we have shown earlier in this paragraph.

For the rest of this proof, assume that f and g are unimodular with $\min\{\operatorname{len} f, \operatorname{len} g\} > 1$. If (f,g) is a Golay pair, then what we have already shown proves that our lower bound is met. Conversely, if our lower bound is met, then we know that there is some nonzero $\lambda \in \mathbb{C}$ such that $(f, |\lambda|g)$ is a Golay pair. Lemma 2.8 shows that $\operatorname{len} f = \operatorname{len}(|\lambda|g)$, so that $\operatorname{len} f = \operatorname{len} g > 1$, and we can use Lemma 2.5(viii) to show that $C_{f,f}(\operatorname{len} f - 1) + C_{|\lambda|g,|\lambda|g}(\operatorname{len} f - 1) = f_{\deg f}\overline{f_{\operatorname{ord} f}} + |\lambda|^2 g_{\deg g}\overline{g_{\operatorname{ord} g}}$, which must equal 0 by the Golay condition. The various coefficients of f and g that appear in the last expression are all unimodular, so this forces $|\lambda| = 1$, and so (f,g) is a Golay pair. So we meet the lower bound precisely when (f,g) is a Golay pair, in which case Lemma 2.10 shows that $\operatorname{len} f = \operatorname{len} g$ and $\mathrm{ADF}(f) = \mathrm{ADF}(g)$. □

Remark 2.11. *Theorem 2.1 is reminiscent of a result of Liu and Guan [**54**, Theorem 1]. Liu and Guan show that binary Golay pairs minimize a criterion that is different from the Pursley–Sarwate criterion in that they do not use the geometric mean of autocorrelation demerit factors, but (if their results are translated into the language of this paper) the arithmetic mean. Liu and Guan also consider the generalization to families that can have more than two sequences, and they generalize demerit factors to weighted versions. For the remainder of this remark, we restrict attention to standard unweighted demerit factors of pairs (as in this paper) of binary sequences of the same length (as in Liu and Guan's paper), so we can compare their results with ours when both results are specialized to those situations where both may be invoked. We translate their results into the notation of this paper: if (f, g) is a pair of binary sequences of equal length, then Liu and Guan's inequality (11) becomes (after a significant amount of calculation)*

$$\frac{\mathrm{CDF}(f, f) + \mathrm{CDF}(f, g) + \mathrm{CDF}(g, f) + \mathrm{CDF}(g, g)}{4} \geq 1,$$

or equivalently, since $\mathrm{CDF}(f, f) = \mathrm{ADF}(f) + 1$, $\mathrm{CDF}(g, g) = \mathrm{ADF}(g) + 1$, *and* $\mathrm{CDF}(g, f) = \mathrm{CDF}(f, g)$,

$$\frac{\mathrm{ADF}(f) + \mathrm{ADF}(g)}{2} + \mathrm{CDF}(f, g) \geq 1. \tag{2.15}$$

Liu and Guan's Theorem 1 asserts that this inequality becomes an exact equality if and only if $\{f, g\}$ is a complementary set. This result is similar to Theorem 2.1, but we note that Liu and Guan's inequality has the arithmetic mean of the autocorrelation demerit factors while Theorem 2.1 has the geometric mean. Theorem 2.1 immediately implies Liu and Guan's result by the arithmetic–geometric mean inequality, but Liu and Guan's result does not immediately imply Theorem 2.1.

2.5. Transformations of Golay Pairs

In this section, we summarize known transformations for Golay pairs and show how they influence the autocorrelation and crosscorrelation behavior of the pairs. In [**26**, p. 496] and [**27**, p. 83], Golay lists six transformations, each of which takes a binary Golay pair (f, g) to another binary Golay pair; these transformations are interchanging the two sequences, reversing the order of the terms in first sequence, reversing the order of the terms in the second sequence, negating the first sequence,

negating the second sequence, and negating every other element in both sequences. To represent these transformations in our Laurent polynomial formalism, if $f(z) \in \mathbb{C}[z, z^{-1}]$, then we define the *conjugate reverse of* $f(z)$, written $f^{\ddagger}(z)$, to be $f^{\ddagger}(z) = z^{\operatorname{ord} f + \deg f} \overline{f(z)}$. Recall our rule that $\infty + (-\infty) = 0$, so that $0^{\ddagger} = 0$. Conjugate reversal preserves degree and order ($\deg f^{\ddagger} = \deg f$ and $\operatorname{ord} f^{\ddagger} = \operatorname{ord} f$), is involutory ($f^{\ddagger\ddagger} = f$), and $|f^{\ddagger}|^2 = |z^{\operatorname{ord} f + \deg f} \overline{f}|^2 = |f|^2$ for every $f \in \mathbb{C}[z, z^{-1}]$. If m, n are integers with $m \leq n$ and $f(z) = f_m z^m + f_{m+1} z^{m+1} + \cdots + f_n z^n \in \mathbb{C}[z, z^{-1}]$ with $f_m, f_n \neq 0$, then $f^{\ddagger}(z) = \overline{f_n} z^m + \overline{f_{n-1}} z^{m+1} + \cdots + \overline{f_m} z^n$, and so conjugate reverse of a contiguous sequence has the same support as the original sequence, but the coefficients are conjugated and arranged in reverse order. We believe that this is the most useful generalization to complex sequences of Golay's transformation that reverses the order of a binary sequence. Borwein and Mossinghoff [**11**, p. 1159] use the reciprocal polynomial $f^*(z) = z^{\deg f} f(1/z)$ to reverse a polynomial $f(z)$ representing a binary sequence, and Katz, Lee, and Trunov (see [**44**, p. 514] and [**45**, p. 7728]) use the conjugate reciprocal polynomial $f^{\dagger}(z)$ (which is obtained from $f^*(z)$ by conjugating every coefficient) as the generalization for polynomials with complex coefficients, but neither of these operations is invertible when one allows sequences whose constant coefficients equal zero, e.g., both z and z^2 have reciprocal (and conjugate reciprocal) equal to 1. Even if one's main interest is in Golay pairs formed from binary sequences represented by polynomials with nonzero constant coefficients, the most comprehensive construction of such Golay pairs, due to Borwein and Ferguson [**12**, Sections 4–5], involves (in intermediate steps) sequences that can have 0 for their constant coefficients. Since we require groups of transformations that work on sequences such as these, the conjugate reversal operation defined here can be used, while the reciprocal and conjugate reciprocal cannot.

If $r(z) \in \mathbb{C}[z, z^{-1}]$, we define the transformation $\operatorname{subs}_r \colon \mathbb{C}[z, z^{-1}]^2 \to \mathbb{C}[z, z^{-1}]^2$ by $\operatorname{subs}_r(f(z), g(z)) = (f(r(z)), g(r(z)))$, that is, subs_r substitutes $r(z)$ for the indeterminate z. We can now define generalizations of Golay's six original transformations.

Definition 2.12 (Elementary Golay transformations)**.** *The elementary Golay transformations are the following transformations that map* $\mathbb{C}[z, z^{-1}]^2$ *to itself.*

(1) $\operatorname{swap}(f, g) = (g, f)$ *(swap the sequences in a pair),*

(2) $\operatorname{conj}(f, g) = (\overline{f}, g)$ *(conjugate the first sequence),*

(3) $\operatorname{conj}'(f, g) = (f, \overline{g})$ *(conjugate the second sequence),*

(4) $\mathrm{crev}(f(z), g(z)) = (f^{\ddagger}, g)$ *(conjugate reverse the first sequence),*

(5) $\mathrm{crev}'(f(z), g(z)) = (f, g^{\ddagger})$ *(conjugate reverse the second sequence),*

(6) $\mathrm{srev}(f(z), g(z))$

$$= (z^{\operatorname{ord} f + \deg f} f(z^{-1}), z^{\operatorname{ord} g + \deg g} g(z^{-1}))$$

(simultaneously reverse the sequences),

(7) $\mathrm{scal}_{p,q}(f, g)) = (pf, qg)$ *(scale by any monomials p and q with $|p|^2 = |q|^2$), and*

(8) $\mathrm{subs}_r(f(z), g(z)) = (f(r(z)), g(r(z)))$ *(substitute a monomial $r(z) = wz^d$ with $|w| = 1$ and $d \in \{-1, 1\}$ for the indeterminate z).*

Golay's six transformations are swap, crev, crev$'$, $\mathrm{scal}_{-1,1}$, $\mathrm{scal}_{1,-1}$, and subs_{-z}.

Lemma 2.13. *Each elementary Golay transformation is a permutation of $\mathbb{C}[z, z^{-1}]^2$ whose inverse is another elementary Golay transformation:* swap, conj, conj$'$, crev, crev$'$, *and* srev *are involutions, the inverse of* $\mathrm{scal}_{p,q}$ *is* $\mathrm{scal}_{1/p,1/q}$*, and the inverse of* subs_{wz^d} *is* $\mathrm{subs}_{w^{-d}z^d}$.

Proof. All of the claims are easy to check, and note that when $p(z)$ and $q(z)$ are monomials with $|p(z)|^2 = |q(z)|^2$, then $1/p(z)$ and $1/q(z)$ are also monomials and have $|1/p(z)|^2 = |1/q(z)|^2$, and if $r(z) = wz^d$ is a monomial with $|w| = 1$ and $d \in \{-1, 1\}$, then $s(z) = w^{-d}z^d$ is a monomial of the same degree with $|w^{-d}| = 1$. □

Definition 2.14 (Golay group, Gol)**.** *The Golay group, written* Gol*, is the group of permutations of $\mathbb{C}[z, z^{-1}]^2$ generated by the elementary Golay transformations.*

We also define another type of transformation of $\mathbb{C}[z, z^{-1}]^2$ that is not, in general, invertible.

Definition 2.15 (Dilation)**.** *For a nonzero integer d, the dilation by d is the transformation* subs_{z^d}.

Definition 2.16 (Extended Golay monoid, EGol)**.** *The extended Golay monoid, written* EGol*, is the monoid of maps from $\mathbb{C}[z, z^{-1}]^2$ to itself generated by the Golay group* Gol *and the dilation maps* $\{\mathrm{subs}_{z^d} : d \in \mathbb{Z}, d \neq 0\}$.

We now examine sets of generators for Gol and EGol.

Lemma 2.17. *Let S be the set whose elements are* swap, conj, crev, srev, $\mathrm{scal}_{p,q}$ *for all monomials p and q with $|p|^2 = |q|^2$, and* subs_r *for all monomials $r = wz^d$ with $|w| = 1$ and $d \in \{-1, 1\}$. Then the monoid generated by S is* Gol*, and therefore the group generated by S is* Gol*. Let T be the set whose elements are* swap, conj, crev, srev, $\mathrm{scal}_{p,q}$ *for all monomials p and q with $|p|^2 = |q|^2$, and* subs_r *for all monomials $r = wz^d$ with $|w| = 1$ and $d \neq 0$. Then the monoid generated by T is* EGol.

Proof. Note that S is the set obtained by removing conj' and crev' from the set of elementary Golay transformations, but since $\mathrm{conj}' = \mathrm{swap} \circ \mathrm{conj} \circ \mathrm{swap}$ and $\mathrm{crev}' = \mathrm{swap} \circ \mathrm{crev} \circ \mathrm{swap}$, the monoid generated by the S includes all the elementary Golay transformations, and since the set of elementary Golay transformations is closed under inversion by Lemma 2.13, the monoid generated by the S is Gol. Every transformation of the form subs_r such that $r(z) = wz^d$ with $|w| = 1$ and $d \neq 0$ is in EGol, since $\mathrm{subs}_r = \mathrm{subs}_{z^d} \circ \mathrm{subs}_{wz}$, which is a composition of a dilation and an element of Gol, and so T is a superset of S but a subset of EGol, so it generates a monoid including Gol and included in EGol. The set T also contains all the dilations, so the monoid it generates must be all of EGol. □

Golay was interested in his six original transformations because they map Golay pairs to Golay pairs. We shall see that the same is true of elements of our extended Golay monoid, after recording without proof some very straightforward principles.

Lemma 2.18. *Let $(a, b) \in \mathbb{C}[z, z^{-1}]^2$ and $\gamma \in$ Gol, and set $(f, g) = \gamma(a, b)$. If $\gamma =$ swap, then $(|a|^2, |b|^2) = (|g|^2, |f|^2)$. If $\gamma \in \{\mathrm{conj}, \mathrm{conj}', \mathrm{crev}, \mathrm{crev}'\}$, then $(|f|^2, |g|^2) = (|a|^2, |b|^2)$. If $\gamma = \mathrm{scal}_{p,q}$ where p and q are monomials with $|p|^2 = |q|^2$, then $|p|^2$ is a positive real number and $(|f|^2, |g|^2) = (|p|^2|a|^2, |p|^2|b|^2)$. If $\gamma =$ srev, then $|f|^2_k = |a|^2_{-k}$ and $|g|^2_k = |b|^2_{-k}$ for every $k \in \mathbb{Z}$. If $\gamma = \mathrm{subs}_r$ where $r = wz^d$ with $|w| = 1$ and $d \neq 0$, then $|f|^2_j = |g|^2_j = 0$ for all $j \in \mathbb{Z}$ with $d \nmid j$, and $|f|^2_{dk} = w^k|a|^2_k$ and $|g|^2_{dk} = w^k|b|^2_k$ for every $k \in \mathbb{Z}$.*

Now we prove that transformations from EGol preserve Golay complementarity.

Lemma 2.19. *If $(a, b) \in \mathbb{C}[z, z^{-1}]^2$ and $\gamma \in$ EGol, then (a, b) is a Golay pair if and only if $\gamma(a, b)$ is a Golay pair.*

PROOF. In view of Lemma 2.17, it suffices to show this in the case where γ is one of swap, conj, crev, srev, $\mathrm{scal}_{p,q}$ (for any monomials p, q with $|p|^2 = |q|^2$), subs_r (for any nonconstant monomial r with a unimodular coefficient). If γ is swap, conj, crev, or $\mathrm{scal}_{p,q}$, and we let $(f, g) = \gamma(a, b)$, then Lemma 2.18 shows that there is some positive real number λ such that $\{|f|^2, |g|^2\} = \{\lambda|a|^2, \lambda|b|^2\}$, and so $|f|^2 + |g|^2 = \lambda(|a|^2 + |b|^2)$ is constant if and only if $|a|^2 + |b|^2$ is constant. If $\gamma = \mathrm{srev}$, then Lemma 2.18 shows that $(|f|^2 + |g|^2)_k = (|a|^2 + |b|^2)_{-k}$ for every $k \in \mathbb{Z}$, so $|f|^2 + |g|^2$ is constant if and only if $|a|^2 + |b|^2$ is constant. If $\gamma = \mathrm{subs}_r$ for $r(z) = wz^d$ with $|w| = 1$ and $d \neq 0$, then Lemma 2.18 shows that $|f|^2 + |g|^2$ only has terms whose degrees are multiples of d, and $(|f|^2 + |g|^2)_{dk} = w^k(|a|^2 + |b|^2)_k$ for every $k \in \mathbb{Z}$, so that $|f|^2 + |g|^2$ is constant if and only if $|a|^2 + |b|^2$ is constant. □

Our transformations also preserve the Pursley–Sarwate criterion.

Lemma 2.20. *Let $(a, b) \in \mathbb{C}[z, z^{-1}]^2$ with $a, b \neq 0$, let $\gamma \in \mathrm{EGol}$, and set $(f, g) = \gamma(a, b)$. Then (f, g) is isoenergetic if and only if (a, b) is isoenergetic. Also $\{\mathrm{ADF}(f), \mathrm{ADF}(g)\} = \{\mathrm{ADF}(a), \mathrm{ADF}(b)\}$ and $\mathrm{CDF}(f, g) = \mathrm{CDF}(a, b)$, and thus $\mathrm{PSC}(f, g) = \mathrm{PSC}(a, b)$.*

PROOF. In view of Lemma 2.17, it suffices to show this in the case where γ is one of swap, conj, crev, srev, $\mathrm{scal}_{p,q}$ (for any monomials p, q with $|p|^2 = |q|^2$), or subs_r (for any nonconstant monomial r with a unimodular coefficient). If γ is swap, conj, crev, or $\mathrm{scal}_{p,q}$, then Lemma 2.18 shows that there is some positive real number λ such that $\{|f|^2, |g|^2\} = \{\lambda|a|^2, \lambda|b|^2\}$. Thus $|f|_0^2 = |g|_0^2$ if and only if $|a|_0^2 = |b|_0^2$, so by (2.7) we see that (f, g) is isoenergetic if and only if (a, b) is. Also $\{|f|_0^4/(|f|_0^2)^2, |g|_0^4/(|g|_0^2)^2\} = \{|a|_0^4/(|a|_0^2)^2, |b|_0^4/(|b|_0^2)^2\}$, and so by (2.10) we have $\{\mathrm{ADF}(f), \mathrm{ADF}(g)\} = \{\mathrm{ADF}(a), \mathrm{ADF}(b)\}$. Similarly, $|fg|_0^2/(|f|_0^2|g|_0^2) = |ab|_0^2/(|a|_0^2|b|_0^2)$, and so by (2.9) we have $\mathrm{CDF}(a, b) = \mathrm{CDF}(f, g)$.

Now assume either that $\gamma = \mathrm{srev}$, in which case

$$(f(z), g(z)) = (z^{\operatorname{ord} a + \deg a} a(z^{-1}), z^{\operatorname{ord} b + \deg b} b(z^{-1})),$$

or that $\gamma = \mathrm{subs}_r$, in which case $(f(z), g(z)) = (a(r(z)), b(r(z)))$. In either case $(|f(z)|^2, |g(z)|^2) = (|a(s(z))|^2, |b(s(z))|^2)$ for some nonconstant monomial $s(z)$ with a unimodular coefficient. Thus, for every $h(z) \in \mathbb{C}[z, z^{-1}]$, the fact that $s(z)$ is a nonconstant monomial with a unimodular coefficient

makes $|h(s(z))|_0^2 = |h|_0^2$. If we consider $h = a$ (resp., $h = b$) and look at (2.7), we see that $\|a\|_2 = \|f\|_2$ (resp., $\|b\|_2 = \|g\|_2$), and so (f, g) is isoenergetic if and only if (a, b) is isoenergetic. On the other hand, if we consider $h \in \{a^2, a\}$ (resp., $h \in \{b^2, b\}$) and look at (2.10), we see that $\mathrm{ADF}(f) = \mathrm{ADF}(a)$ (resp., $\mathrm{ADF}(g) = \mathrm{ADF}(b)$), and if we consider $h \in \{ab, a, b\}$ and look at (2.9), we see that $\mathrm{CDF}(f, g) = \mathrm{CDF}(a, b)$. □

Sometimes we are only interested in the transformations that preserve certain properties of pairs, so we define some special subgroups of Gol.

Definition 2.21 (Stationary Golay Group, SGol)**.** *The stationary Golay group, written* SGol*, is the subgroup of* Gol *generated by* swap*,* crev*,* crev′*,* srev*, all transformations* $\mathrm{scal}_{u,v}$ *where* u, v *are nonzero complex numbers with* $|u| = |v|$*, and all transformations* subs_{wz}*, where* w *is a unimodular complex number.*

Remark 2.22. *Suppose that* $(a, b) \in \mathbb{C}[z, z^{-1}]^2$*,* $\gamma \in$ SGol*, and* $(f, g) = \gamma(a, b)$*. Then one easily sees that*

$$\{(\operatorname{ord} f, \deg f), (\operatorname{ord} g, \deg g)\} = \{(\operatorname{ord} a, \deg a), (\operatorname{ord} b, \deg b)\}$$

since all the generators and inverses of generators of SGol *except for* swap *preserve the orders and degrees of both sequences in the pair, while* swap *swaps the two sequences and the pair. In particular, transformations in* SGol *preserve the minimum and maximum values of order (or of degree, or of length) for the two sequences in the pair. Therefore, if* $\operatorname{ord} a = \operatorname{ord} b$ *(resp.,* $\deg a = \deg b$*,* $\operatorname{len} a = \operatorname{len} b$*), then* $\mathrm{ord}(f, g) = \mathrm{ord}(a, b)$ *(resp.,* $\deg(f, g) = \deg(a, b)$*,* $\mathrm{len}(f, g) = \mathrm{len}(a, b)$*).*

Definition 2.23 (Unimodular Golay Group UGol)**.** *The unimodular Golay group, written* UGol*, is the subgroup of* Gol *generated by* swap*,* crev*,* crev′*,* srev*, all transformations* $\mathrm{scal}_{u,v}$ *where* u, v *are unimodular complex numbers, and all transformations* subs_{wz} *where* w *is a unimodular complex number.*

Remark 2.24. *If* $(f, g) \in \mathbb{C}[z, z^{-1}]$ *and* $\gamma \in$ UGol*, then one readily sees* (f, g) *is unimodular if and only if* $\gamma(f, g)$ *is unimodular. Also,* UGol *is clearly a subgroup of* SGol*, and so it has the properties mentioned in Remark 2.22.*

Definition 2.25 (Binary Golay Group BGol)**.** *The binary Golay group, written* BGol*, is the subgroup of* Gol *generated by* swap*,* crev*,* crev′*,* srev*,* $\text{scal}_{-1,1}$*,* $\text{scal}_{1,-1}$*, and* subs_{-z}*.*

Remark 2.26. *The binary Golay group is simply the group generated by Golay's six transformations along with* srev*, but one should note that if we are applying transformations in* BGol *to sequences with real terms, then* srev *has the same effect as* $\text{crev} \circ \text{crev}'$*. Clearly* BGol *is a subgroup of* UGol*, since it is generated by a subset of* UGol*'s generators. If* $(f, g) \in \mathbb{C}[z, z^{-1}]$ *and* $\gamma \in$ BGol*, then one readily sees* (f, g) *is binary if and only if* $\gamma(f, g)$ *is binary. Also,* BGol *is clearly a subgroup both of* SGol *and of* UGol*, and so it has the properties mentioned in Remarks 2.22 and 2.24.*

2.6. Constructions of Golay Pairs

In this section we describe construction methods, each of which takes two Golay pairs, (a, b) and (c, d), and produces another Golay pair (f, g). Golay, Turyn, and Borwein–Ferguson gave various methods of this kind. We devise a single simple method which, using the transformations from the extended Golay monoid and the pair $(1, 1)$ (which is obviously Golay), can construct all Golay pairs obtainable by these earlier methods.

Construction 2.27 (Weaving)**.** *Let* $a, b, c, d \in \mathbb{C}[z, z^{-1}]$*. Then the weave of* (a, b) *with* (c, d)*, denoted* $(a, b) \ltimes (c, d)$ *or* $(c, d) \rtimes (a, b)$*, is the pair* (f, g)*, where*

$$f(z) = a(z)c(z) + b(z)d(z)$$
$$g(z) = a(z)\overline{d(z)} - b(z)\overline{c(z)}.$$

We want to show that this construction produces Golay pairs when the inputs are Golay pairs.

Lemma 2.28. *Let $a, b, c, d \in \mathbb{C}[z, z^{-1}]$ and let $(f, g) = (a, b) \ltimes (c, d)$. Then*

$$|f|^2 = |ac|^2 + |bd|^2 + 2\operatorname{Re}(ac\overline{bd})$$

$$|g|^2 = |ad|^2 + |bc|^2 - 2\operatorname{Re}(ac\overline{bd}), \text{ and}$$

$$|f|^2 + |g|^2 = (|a|^2 + |b|^2)(|c|^2 + |d|^2).$$

PROOF. Since $f = ac + bd$, we have $|f|^2 = |ac|^2 + |bd|^2 + ac\overline{bd} + \overline{ac}bd$, which verifies the first formula. And since $g = a\overline{d} - b\overline{c}$, we have $|g|^2 = |ad|^2 + |bc|^2 - ac\overline{bd} - \overline{ac}bd$, which verifies the second formula. And then

$$\begin{aligned} &|f|^2 + |g|^2 \\ &= |ac|^2 + |bd|^2 + 2\operatorname{Re}(ac\overline{bd}) + |ad|^2 + |bc|^2 - 2\operatorname{Re}(ac\overline{bd}) \\ &= (|a|^2 + |b|^2)(|c|^2 + |d|^2). \end{aligned}$$ □

Corollary 2.29. *Let $a, b, c, d \in \mathbb{C}[z, z^{-1}]$. If both (a, b) and (c, d) are not equal to $(0, 0)$, then $(a, b) \ltimes (c, d)$ is a Golay pair if and only if both (a, b) and (c, d) are Golay pairs. If $(a, b) = (0, 0)$ or $(c, d) = (0, 0)$, then $(a, b) \ltimes (c, d) = (0, 0)$, which is a Golay pair.*

PROOF. The claim when (a, b) or (c, d) is $(0, 0)$ is clear, so suppose neither is $(0, 0)$, and let $(f, g) = (a, b) \ltimes (c, d)$. Lemma 2.28 shows that $|f|^2 + |g|^2$ is constant if both $|a|^2 + |b|^2$ and $|c|^2 + |d|^2$ are. By (2.7) and the fact that $(a, b), (c, d) \neq (0, 0)$, we see that $|a|^2 + |b|^2$ and $|c|^2 + |d|^2$ have nonzero constant terms. If either of them were nonconstant, then their product $|f|^2 + |g|^2$ would have more than one monomial, and hence be nonconstant. So $|f|^2 + |g|^2$ is a constant if and only if both $|a|^2 + |b|^2$ and $|c|^2 + |d|^2$ are, so the result follows from Lemma 2.7. □

The following technical lemma, which will be used later, considers how the weaving construction influences expressions that appear in our Laurent polynomial formulae (2.10) and (2.9) for autocorrelation and crosscorrelation demerit factors.

Lemma 2.30. *If $a, b, c, d \in \mathbb{C}[z, z^{-1}]$ and $(f, g) = (a, b) \ltimes (c, d)$, then we have*

$$\begin{aligned}|f|^4 &= |ac|^4 + |bd|^4 + 4|abcd|^2 \\ &\quad + 4(|ac|^2 + |bd|^2)\operatorname{Re}(ac\overline{bd}) + 2\operatorname{Re}((ac\overline{bd})^2), \\ |g|^4 &= |ad|^4 + |bc|^4 + 4|abcd|^2 \\ &\quad - 4(|ad|^2 + |bc|^2)\operatorname{Re}(ac\overline{bd}) + 2\operatorname{Re}((ac\overline{bd})^2),\end{aligned}$$

and

$$\begin{aligned}|fg|^2 = |ab|^2(|c|^4 + |d|^4) + |cd|^2(|a|^4 + |b|^4) - 2|abcd|^2 \\ - 2(|a|^2 - |b|^2)(|c|^2 - |d|^2)\operatorname{Re}(ac\overline{bd}) - 2\operatorname{Re}((ac\overline{bd})^2).\end{aligned}$$

Proof. By Lemma 2.28, we have $|f|^2 = |ac|^2 + |bd|^2 + 2\operatorname{Re}(ac\overline{bd})$. Square this and use the identity $4\operatorname{Re}(u)^2 = 2\operatorname{Re}(u^2) + 2|u|^2$ to obtain the result for $|f|^4$. One obtains the result for $|g|^4$ the same way by replacing c and d with $\overline{d}$ and $-\overline{c}$, respectively. From our expressions for $|f|^4$ and $|g|^4$, we have

$$\begin{aligned}|f|^4 + |g|^4 = (|a|^4 + |b|^4)(|c|^4 + |d|^4) + 8|abcd|^2 \\ + 4(|a| - |b|^2)(|c|^2 - |d|^2)\operatorname{Re}(ac\overline{bd}) + 4\operatorname{Re}((ac\overline{bd})^2).\end{aligned}$$

Then note that

$$|fg|^2 = \frac{(|f|^2 + |g|^2)^2 - (|f|^4 + |g|^4)}{2},$$

and then we use the expression for $|f|^2 + |g|^2$ from Lemma 2.28 and our expression for $|f|^4 + |g|^4$ to deduce the expression for $|fg|^2$. □

Now we describe the historical methods that were used to construct binary Golay pairs. We present them in slightly more general forms than the originals so as to make them also suitable for constructing Golay pairs with complex terms. The first two constructions are due to Golay: see [**27**, eqs. (10),(11)].

Construction 2.31 (Golay concatenation)**.** *If* (a, b) *and* (c, d) *are Golay pairs with* $\operatorname{ord} c + \deg c = \operatorname{ord} d + \deg d$*, and* m *and* n *are integers with* $m \neq 0$*, and we let* $f(z) = a(z)c(z^m) + z^{mn}b(z)d(z^m)$ *and* $g(z) = a(z)d^\ddagger(z^m) - z^{mn}b(z)c^\ddagger(z^m)$*, then* (f, g) *is*

$$\operatorname{scal}_{1,z^{m(\operatorname{ord} c+\deg c)}}\left((\operatorname{scal}_{1,z^{mn}}(a, b)) \ltimes (\operatorname{subs}_{z^m}(c, d))\right),$$

which is a Golay pair. In particular, if (a, b) *and* (c, d) *are unimodular (resp., binary) Golay pairs with* $\operatorname{len}(a, b) = m$*,* $\operatorname{len}(c, d) = n$*,* $\operatorname{ord} a = \operatorname{ord} b$*, and* $\operatorname{ord} c = \operatorname{ord} d$*, then* (f, g) *is a unimodular (resp., binary) Golay pair with* $\operatorname{len}(f, g) = 2mn$*.*

Proof. It is not difficult to check the formula for (f, g), from which it follows that (f, g) is Golay via Lemma 2.19 and Corollary 2.29. The second claim is clear if (a, b) or (c, d) is $(0, 0)$, and otherwise follows from the fact that the coefficient of z^j in $f(z)$ (resp., $g(z)$) is 0 unless $\operatorname{ord} a + m \operatorname{ord} c \leq j \leq mn + \deg a + m \deg c$, in which case it is the either the product of a nonzero coefficient of a with one of c or the product of a nonzero coefficient of b with one of d (resp., either the product of a nonzero coefficient of a with the conjugate of one of d or the product of a nonzero coefficient of b with the conjugate of one of c). □

Construction 2.32 (Golay interleaving [**27**])**.** *If* (a, b) *and* (c, d) *are Golay pairs with* $\operatorname{ord} c + \deg c = \operatorname{ord} d + \deg d$*, and* m *is a nonzero integer, and we let* $f(z) = a(z)c(z^{2m}) + z^m b(z)d(z^{2m})$ *and* $g(z) = a(z)d^\ddagger(z^{2m}) - z^m b(z)c^\ddagger(z^{2m})$*, then* (f, g) *is*

$$\operatorname{scal}_{1,z^{2m(\operatorname{ord} c+\deg c)}}\left((\operatorname{scal}_{1,z^m}(a, b)) \ltimes (\operatorname{subs}_{z^{2m}}(c, d))\right),$$

which is a Golay pair. In particular, if (a, b) *and* (c, d) *are unimodular (resp., binary) Golay pairs with* $\operatorname{len}(a, b) = m$*,* $\operatorname{len}(c, d) = n$*,* $\operatorname{ord} a = \operatorname{ord} b$*, and* $\operatorname{ord} c = \operatorname{ord} d$*, then* (f, g) *is a unimodular (resp., binary) Golay pair with* $\operatorname{len}(f, g) = 2mn$*.*

Proof. It is not difficult to check the formula for (f, g), from which it follows that (f, g) is Golay via Lemma 2.19 and Corollary 2.29. The second claim is clear if (a, b) or (c, d) is $(0, 0)$, and otherwise follows from the fact that the coefficient of z^j in $f(z)$ (resp., $g(z)$) is 0 unless $\operatorname{ord} a + 2m \operatorname{ord} c \leq j \leq m + \deg a + 2m \deg c$, in which case it is the either the product of a nonzero coefficient of a with one of c or the product of a nonzero coefficient of b with one of d (resp., either

the product of a nonzero coefficient of a with the conjugate of one of d or the product of a nonzero coefficient of b with the conjugate of one of c). □

The next construction was presented by Borwein and Ferguson [**12**, p. 975] as being Golay's interleaving construction. In fact, it is a slightly different (but still valid) way of combining two Golay pairs to get a longer one.

Construction 2.33 (Borwein-Ferguson interleaving [**12**])**.** *If (a, b) and (c, d) are Golay pairs with $\operatorname{ord} c + \deg c = \operatorname{ord} d + \deg d$ and m is a nonzero integer, and we let $f(z) = a(z^2)c(z^{2m}) + zb(z^2)d(z^{2m})$ and let $g(z) = a(z^2)d^\ddagger(z^{2m}) - zb(z^2)c^\ddagger(z^{2m})$, then (f, g) is*

$$\operatorname{scal}_{1,z^{2m(\operatorname{ord} c+\deg c)}}\left((\operatorname{scal}_{1,z}\operatorname{subs}_{z^2}(a, b)) \ltimes (\operatorname{subs}_{z^{2m}}(c, d))\right),$$

which is a Golay pair. In particular, if (a, b) and (c, d) are unimodular (resp., binary) Golay pairs with $\operatorname{len}(a, b) = m$, $\operatorname{len}(c, d) = n$, $\operatorname{ord} a = \operatorname{ord} b$, and $\operatorname{ord} c = \operatorname{ord} d$, then (f, g) is a unimodular (resp., binary) Golay pair with $\operatorname{len}(f, g) = 2mn$.

Proof. It is not difficult to check the formula for (f, g), from which it follows that (f, g) is Golay via Lemma 2.19 and Corollary 2.29. The second claim is clear if (a, b) or (c, d) is $(0, 0)$, and otherwise follows from the fact that the coefficient of z^j in $f(z)$ (resp., $g(z)$) is 0 unless $2 \operatorname{ord} a + 2m \operatorname{ord} c \leq 1 + 2 \deg a + 2m \deg c$, in which case it is the either the product of a nonzero coefficient of a with one of c or the product of a nonzero coefficient of b with one of d (resp., either the product of a nonzero coefficient of a with the conjugate of one of d or the product of a nonzero coefficient of b with the conjugate of one of c). □

The final construction is due to Turyn [**92**, Lemma 5].[a]

Construction 2.34 (Turyn)**.** *If (a, b) and (c, d) are Golay pairs with $\operatorname{ord} a + \deg a = \operatorname{ord} b + \deg b$ and m is a nonzero integer, and we let*

$$f(z) = a(z)\left(\frac{c(z^m) + d(z^m)}{2}\right) - b^\ddagger(z)\left(\frac{c(z^m) - d(z^m)}{2}\right)$$

[a]Note that Turyn uses the notation $A \times B$ to denote the tensor product $B \otimes A$; this is clear when one compares his summaries (immediately preceding Lemma 5) of Golay's constructions with the originals in [**27**, eq. (10),(11)].

and

$$g(z) = b(z)\left(\frac{c(z^m) + d(z^m)}{2}\right) + a^{\ddagger}(z)\left(\frac{c(z^m) - d(z^m)}{2}\right),$$

then (f, g) *is*

$$\operatorname{scal}_{1/2,-z^{\operatorname{ord} a+\deg a}/2}\Big(\Big(\operatorname{subs}_{z^m}\big((c,d) \ltimes (1,1)\big)\Big) \ltimes \big(\operatorname{scal}_{1,-1}\operatorname{crev}'(a,b)\big)\Big),$$

which is a Golay pair. In particular, if (a, b) *is a unimodular (resp., binary) Golay pair and* (c, d) *is a binary Golay pair, and if* $\operatorname{len}(a, b) = m$, $\operatorname{len}(c, d) = n$, $\operatorname{ord} a = \operatorname{ord} b$, *and* $\operatorname{ord} c = \operatorname{ord} d$, *then* (f, g) *is a unimodular (resp., binary) Golay pair with* $\operatorname{len}(f, g) = mn$.

Proof. It is not difficult to check the formula for (f, g), from which it follows that (f, g) is Golay via Lemma 2.19, Corollary 2.29, and the fact that $(1, 1)$ is a Golay pair. The second claim is clear if (a, b) or (c, d) is $(0, 0)$. Otherwise, we first note that $(c + d)/2$ and $(c - d)/2$ are sequences whose coefficients for z^j both vanish unless $\operatorname{ord} c \le j \le \deg c$, in which case precisely one of them vanishes and the other is in $\{-1, 1\}$. Thus the coefficient of z^j in $f(z)$ (resp., in $g(z)$) is 0 unless $\operatorname{ord} a + m \operatorname{ord} c \le j \le \deg a + m \deg c$, in which case it is equal to plus or minus either a nonzero coefficient of a or else the conjugate one of b (resp., plus or minus either a nonzero coefficient of b or else the conjugate of one of a). □

For arbitrary nonnegative integers a, b, and c, one can construct a binary Golay pair of length $2^a 10^b 26^c$ using Turyn's construction and the following sequence pairs (represented using + to denote $+1$ and − to denote -1).

- Length 1:

  \+
  \+

- Length 2:

  \+ \+
  \+ −

- Length 10:

  \+\+\+\+\+−\+−−\+
  \+\+−−\+\+\+−\+−

or

++−++++−−
++−+−+−−++

- Length 26:

++++−++−−+−+++++−+−−−++−−−
++++−++−−+−+−+−−+−+++−−+++

We have listed two Golay pairs of length 10 that are inequivalent modulo the action of the binary Golay group BGol (all other binary Golay pairs of length 10 are equivalent to one of these); for the other lengths listed above there is only one binary Golay pair of that length modulo the action of BGol (see [**19**, Table 1]). We note that each sequence in the second Golay pair of length 10 above can be obtained from the corresponding sequence in the first Golay pair of length 10 by selecting the next-to-leftmost term and then selecting every third term, proceeding cyclically (cf. [**27**, p. 86]).

Đoković [**19**, Definition 2.1] defines a *constructible* binary Golay pair to be one that is equivalent, modulo the action of the binary Golay group BGol, to a binary Golay pair that can be generated from two smaller binary Golay pairs via a variant of Turyn's construction. In addition to the binary Golay pairs of lengths 1, 2, 10, and 26 displayed above, Đoković shows that (up to equivalence modulo BGol) there are two non-constructible binary Golay pairs of length 16, one of length 20, and forty-four of length 32. Borwein and Ferguson [**12**, pp. 975–978] devise a procedure that produces all binary Golay pairs of length less than 100 starting from a set of only five starting binary Golay pairs. Their procedure, based on Constructions 2.31 and 2.33 and transformations from BGol, allows for non-binary Golay pairs whose terms are in $\{1, -1, 0\}$ in intermediate steps, and their starting binary Golay pairs are the pairs of lengths 1, 10, and 26 listed above, along with the following pair of length 20.

- Length 20:

++++−+++++−−−+−+−++−
++++−+−−−++−−++−+−−+

Since Borwein and Ferguson obtain every binary Golay pair of length less than 100 starting from the specific Golay pairs of lengths 1, 10, 20, and 26 listed above by means of the transformations in BGol along with Constructions 2.31 and 2.33, the fact that these constructions can be obtained from the weaving construction and transformations in EGol shows that every binary Golay pair of length

100 can be obtained from Borwein and Ferguson's five starting Golay pairs, the transformations in EGol, and the weaving construction.

2.7. Iterated Golay-Rudin-Shapiro Construction

In this section we show how the demerit factors of sequences in Golay pairs change when we repeatedly apply a simple construction that produces longer and longer Golay pairs. Before Golay presented Construction 2.31, he had already devised [**26**, p. 469] the special case of it where the second pair used in the construction is $(c, d) = (1, 1)$ and the parameter n is 1. At almost the same time, Shapiro [**80**, pp. 39–40] devised the same special case of Construction 2.31 in his studies of Littlewood polynomials with small L^∞ norm on the complex unit circle, and this was later rediscovered by Rudin [**75**, eq. (1.5)]; for this reason we call this special case of Construction 2.31 the Golay–Rudin–Shapiro construction.

Construction 2.35 (Golay-Rudin-Shapiro)**.** *If $(a, b) \in \mathbb{C}[z, z^{-1}]^2$ and m is an integer, then we set*

$$\begin{aligned}\mathrm{GRS}((a, b), m) &= (\mathrm{scal}_{1,z^m}(a, b)) \ltimes (1, 1)\\ &= (a + z^m b, a - z^m b).\end{aligned}$$

We prove some properties of our construction.

(i). If (a, b) is a Golay pair, then so is $\mathrm{GRS}((a, b), m)$.

(ii). If $\operatorname{ord} a = \operatorname{ord} b$ and $m > 0$, then $\mathrm{GRS}((a, b), m)$ is a pair of order $\operatorname{ord} a$.

(iii). If $\operatorname{len} a = \operatorname{len} b = m$ and $\operatorname{ord} a = \operatorname{ord} b$, then $\mathrm{GRS}((a, b), m)$ is a pair of length $2m$ and order $\operatorname{ord} a$.

(iv). If (a, b) is a unimodular (resp., binary) pair with $\operatorname{len} a = \operatorname{len} b = m$ and $\operatorname{ord} a = \operatorname{ord} b$, then $\mathrm{GRS}((a, b), m)$ is a unimodular (resp., binary) pair of length $2m$ and order $\operatorname{ord} a$.

Proof. The claim that $\mathrm{GRS}((a, b), m)$ is a Golay pair whenever (a, b) is follows from Lemma 2.19 and Corollary 2.29 and the fact that $(1, 1)$ is a Golay pair, and the other claims are clear from the second expression for $\mathrm{GRS}((a, b), m)$. □

We now iterate the previous construction, also allowing for the application of transformations from EGol.

Construction 2.36 (Iterated Golay-Rudin-Shapiro)**.** *Let* $(a, b) \in \mathbb{C}[z, z^{-1}]^2$*, let* $m \in \mathbb{Z}$*, and let* $\gamma = (\gamma_0, \gamma_1, \ldots)$ *be a sequence of transformations from* EGol*. For* $n \in \mathbb{N}$*, we define* $\mathrm{GRS}_\gamma^n((a, b), m)$ *recursively by setting* $\mathrm{GRS}_\gamma^0((a, b), m) = \gamma_0(a, b)$ *and for* $n > 0$ *setting*

$$\mathrm{GRS}_\gamma^{n+1} = \gamma_{n+1}(\mathrm{GRS}(\mathrm{GRS}_\gamma^n((a, b), m), 2^n m)).$$

We prove some properties of our iterated construction.

(i). If (a, b) *is a Golay pair, then so is* $\mathrm{GRS}_\gamma^n((a, b), m)$ *for every* $n \in \mathbb{N}$*.*

(ii). If $\operatorname{ord} a = \operatorname{ord} b$ *and* $m > 0$*, and* $\gamma_j \in \mathrm{SGol}$ *for every* $j \in \mathbb{N}$*, then* $\mathrm{GRS}_\gamma^n((a, b), m)$ *is a pair of order* $\operatorname{ord} a$ *for each* $n \in \mathbb{N}$*.*

(iii). If $\operatorname{len} a = \operatorname{len} b = m$*,* $\operatorname{ord} a = \operatorname{ord} b$*, and* $\gamma_j \in \mathrm{SGol}$ *for every* $j \in \mathbb{N}$*, then* $\mathrm{GRS}_\gamma^n((a, b), m)$ *is a pair of order* $\operatorname{ord} a$ *and length* $2^n m$ *for each* $n \in \mathbb{N}$*.*

(iv). If (a, b) *is unimodular (resp., binary),* $\operatorname{len} a = \operatorname{len} b = m$*,* $\operatorname{ord} a = \operatorname{ord} b$*, and* $\gamma_j \in \mathrm{UGol}$ *(resp.,* BGol*) for every* $j \in \mathbb{N}$*, then* $\mathrm{GRS}_\gamma^n((a, b), m)$ *is a unimodular (resp., binary) pair of order* $\operatorname{ord} a$ *and length* $2^n m$ *for each* $n \in \mathbb{N}$*.*

Proof. Each claim is proved inductively using Construction 2.35 along with Lemma 2.19 (for the first claim), Remark 2.22 (for the second and third claims), and Remark 2.24 (resp., Remark 2.26) for the parts of the fourth claim not already established in the third claim. □

Example 2.37. *We consider some examples of how Construction 2.36 can be used to construct Golay pairs. To easily encode Golay pairs, we represent each binary sequence* f *of length* 2^n *(for* $n \in \mathbb{N}$*) as a Boolean function in* n *variables. This is a very convenient formalism developed by Davis and Jedwab* ***[14]*** *that captures all the known Golay pairs whose lengths are powers of* 2*; these pairs were already known to Golay* ***[27****, pp. 85–86], who obtained them via a closely related formalism. We use the same idea as these previous authors, but with a slightly more convenient indexing. We let* $\mathbb{F}_2$ *be the binary field, and write a Boolean function from* $\mathbb{F}_2^n$ *to* $\mathbb{F}_2$ *as a polynomial* $F(x_0, x_1, \ldots, x_{n-1}) \in \mathbb{F}_2[x_0, x_1, \ldots, x_{n-1}]$*, where* $x_0, x_1, \ldots, x_{n-1}$ *are* n *indeterminates, none of which appears with a power greater than* 1 *in our polynomial* $F(x_0, x_1, \ldots, x_{n-1})$*. We use the convention that if we evaluate* F *with inputs from* $\mathbb{Z}$*, we mean that those elements should be reduced modulo* 2 *(and so mapped into* $\mathbb{F}_2$*) before evaluation. With this convention, the sequence*

$f = \sum_{j \in \mathbb{Z}} f_j z^j$ associated to F has $f_j = (-1)^{F(j_0, j_1, \ldots, j_{n-1})}$ whenever $j_0, \ldots, j_{n-1} \in \{0, 1\} \subseteq \mathbb{Z}$ with $j = \sum_{u=0}^{n-1} j_u 2^u$ (and $f_j = 0$ when $j \notin \{0, 1, \ldots, 2^n - 1\}$). Note this concept even works for the the sequences 1 and -1 of length 1: these correspond to the Boolean functions in zero variables (i.e., constants) $F = 0$ and $G = 1$, respectively. Note that negation (resp., substitution of $-z$ for z, conjugate reversal) of the binary sequence of length 2^n associated with Boolean function $F(x_0, \ldots, x_{n-1})$ changes it into the binary sequence of length 2^n associated with Boolean function $F(x_0, \ldots, x_{n-1}) + 1$ (resp., $F(x_0, \ldots, x_{n-1}) + x_0$, $F(x_0 + 1, \ldots, x_{n-1} + 1)$).

If (a, b) is a binary Golay pair of length 2^n whose sequences correspond to Boolean functions $A(x_0, \ldots, x_{n-1})$ and $B(x_0, x_1, \ldots, x_{n-1})$, respectively, then it is not hard to show that $\mathrm{GRS}((a, b), 2^n)$ is the binary Golay pair of length 2^{n+1} whose sequences correspond to the Boolean functions

$$(1 - x_n)A(x_0, \ldots, x_{n-1}) + x_n B(x_0, \ldots, x_{n-1})$$

and

$$(1 - x_n)A(x_0, \ldots, x_{n-1}) + x_n B(x_0, \ldots, x_{n-1}) + x_n,$$

respectively. Therefore, if we start with (a, b) equal to the binary Golay pair $(1, 1)$ of length 1, and if we let $\gamma = (\gamma_0, \gamma_1, \ldots)$ where every γ_j is the identity transformation, then it is not hard to use induction to show that $\mathrm{GRS}_\gamma^n((a, b), 1)$ is the binary Golay pair of length 2^n whose sequences correspond to the Boolean functions 0 and 0 (if $n = 0$) or

$$\sum_{j=0}^{n-2} x_j x_{j+1} \text{ and } \sum_{j=0}^{n-2} x_j x_{j+1} + x_{n-1}$$

(if $n \geq 1$), where we construe empty sums as 0 when $n = 1$. These are the Boolean functions associated to the original sequence pairs of Golay and Shapiro.

On the other hand, we can obtain very different sequences starting from the same initial pair $(1, 1)$ if we use some non-identity transformations between the stages of Golay concatenation. Now we let γ_j be the identity map for all even j, but let $\gamma_j = \mathrm{crev}$ for all odd j. Then it is not hard to show by induction that if (a, b) is the binary Golay pair $(1, 1)$ of length 1, then $\mathrm{GRS}_\gamma^n((a, b), 1)$ is the binary Golay pair of length 2^n associated to the Boolean functions 0 and 0 (for $n = 0$), 0 and x_0 (for

$n = 1$), or $C(x_0, \ldots, x_{n-1})$ and $D(x_0, \ldots, x_{n-1})$ (for $n \geq 2$), where

$$C(x_0, \ldots, x_{n-1}) = x_0 x_1 + \sum_{j=0}^{\lfloor (n-3)/2 \rfloor} x_{2j+1} x_{2j+2} + \sum_{j=0}^{\lfloor (n-4)/2 \rfloor} x_{2j} x_{2j+3} + x_0 + x_{2\lfloor (n-1)/2 \rfloor},$$

and

$$D(x_0, \ldots, x_{n-1}) = C(x_0, \ldots, x_{n-1}) + x_{n-2+(-1)^n},$$

and we construe empty sums as 0 *when* n *is small.*

We are interested in the autocorrelation and crosscorrelation demerit factors of $\mathrm{GRS}_\gamma^n((a, b), m)$ as a function of the inputs. We begin with a useful technical result.

Lemma 2.38. *Let* (a, b) *be an isoenergetic Golay pair with* $a, b \neq 0$. *Then*

$$\begin{aligned} \frac{|a|_0^4 + |b|_0^4}{(|a|_0^2 + |b|_0^2)^2} &= \frac{\mathrm{ADF}(a) + 1}{2} \\ &= \frac{\mathrm{ADF}(b) + 1}{2} \\ &= 1 - \frac{\mathrm{CDF}(a, b)}{2}. \end{aligned}$$

Proof. Since (a, b) is isoenergetic we have $|a|_0^2 = |b|_0^2$, and Lemma 2.9 shows that we have $\mathrm{ADF}(a) = \mathrm{ADF}(b)$, and so

$$\begin{aligned} \frac{|a|_0^4 + |b|_0^4}{(|a|_0^2 + |b|_0^2)^2} &= \frac{|a|_0^4}{4(|a|_0^2)^2} + \frac{|b|_0^4}{4(|b|_0^2)^2} \\ &= \frac{\mathrm{ADF}(a) + 1}{4} + \frac{\mathrm{ADF}(b) + 1}{4} \\ &= \frac{\mathrm{ADF}(a) + 1}{2}, \end{aligned}$$

where we used (2.10) in the second equality. Since (a, b) is a Golay pair with $\mathrm{ADF}(a) = \mathrm{ADF}(b)$, Theorem 2.1 shows that $\mathrm{ADF}(a) = 1 - \mathrm{CDF}(a, b)$, and substituting this in our expression completes the proof. □

Now we show the effect of one step of Construction 2.36 on the demerit factors.

Proposition 2.39. *Let* (a, b) *be an isoenergetic Golay pair with* $\operatorname{ord} a = \operatorname{ord} b \neq \infty$, *and let* m *be an integer with* $\max(\operatorname{len} a, \operatorname{len} b) \leq m$. *Let* $\sigma \in \mathrm{SGol}$. *If* $(f, g) = \sigma(\mathrm{GRS}((a, b), m))$, *then* (f, g) *is an*

isoenergetic Golay pair with $\operatorname{ord}(f,g) = \operatorname{ord}(a,b) \neq \infty$, $\max(\operatorname{len} f, \operatorname{len} g) \leq 2m$, $\operatorname{ADF}(f) = \operatorname{ADF}(g)$, *and*

$$\operatorname{ADF}(f) - \frac{1}{3} = -\frac{1}{2}\left(\operatorname{ADF}(a) - \frac{1}{3}\right)$$
$$\operatorname{CDF}(f,g) - \frac{2}{3} = -\frac{1}{2}\left(\operatorname{CDF}(a,b) - \frac{2}{3}\right).$$

Proof. First of all, note that $m > 0$ since the condition that $\operatorname{ord}(a,b) \neq \infty$ gives a and b positive lengths. Let $(c,d) = \operatorname{GRS}((a,b),m)$, so that $(f,g) = \sigma(c,d)$. From Construction 2.35 and we know that (c,d) is a Golay pair with $\operatorname{ord}(c,d) = \operatorname{ord}(a,b) \neq \infty$, and then we can see that $(c,d) = (a + z^m b, a - z^m b)$ has $\max(\operatorname{len} c, \operatorname{len} d) \leq 2m$. Then by Lemma 2.19 and Remark 2.22, we see that (f,g) is a Golay pair with $\operatorname{ord}(f,g) = \operatorname{ord}(a,b) \neq \infty$ and $\max(\operatorname{len} f, \operatorname{len} g) \leq 2m$.

Since $(c,d) = \operatorname{GRS}((a,b),m) = (a, z^m b) \ltimes (1,1)$, we apply Lemma 2.28 and Lemma 2.30 and extract the constant coefficients to obtain

$$|c|_0^2 = |a|_0^2 + |b|_0^2 + 2\operatorname{Re}(z^{-m} a\bar{b})_0$$
$$|d|_0^2 = |a|_0^2 + |b|_0^2 - 2\operatorname{Re}(z^{-m} a\bar{b})_0$$
$$|cd|_0^2 = |a|_0^4 + |b|_0^4 - 2\operatorname{Re}((z^{-m} a\bar{b})^2)_0.$$

Since a and b are of length m or less and $\operatorname{ord} a = \operatorname{ord} b$, only monomials of negative degree occur in $z^{-m} a\bar{b}$, so there is no constant term in $\operatorname{Re}((z^{-m} a\bar{b})^k)$ for any $k > 0$. Thus we have

$$|c|_0^2 = |a|_0^2 + |b|_0^2$$
$$|d|_0^2 = |a|_0^2 + |b|_0^2$$
$$|cd|_0^2 = |a|_0^4 + |b|_0^4.$$

In view of (2.7), the first two equations show that (c,d) is isoenergetic, so Lemma 2.9 shows that $\operatorname{ADF}(c) = \operatorname{ADF}(d)$. Furthermore we use (2.9) and the expressions above with Lemma 2.38 to see

that

$$\begin{aligned}\mathrm{CDF}(c,d) &= \frac{|a|_0^4 + |b|_0^4}{(|a|_0^2 + |b|_0^2)^2} \\ &= 1 - \frac{\mathrm{CDF}(a,b)}{2},\end{aligned}$$

which means that

$$\mathrm{CDF}(c,d) - \frac{2}{3} = -\frac{1}{2}\left(\mathrm{CDF}(a,b) - \frac{2}{3}\right).$$

Since (c,d) is a Golay pair with $\mathrm{ADF}(c) = \mathrm{ADF}(d)$, Theorem 2.1 tells us that $\mathrm{CDF}(c,d)+\mathrm{ADF}(c) = 1$, and similarly $\mathrm{CDF}(a,b) + \mathrm{ADF}(a) = 1$ because $\mathrm{ADF}(a) = \mathrm{ADF}(b)$ by Lemma 2.9, so we can negate both sides of the last equation to obtain

$$\mathrm{ADF}(c) - \frac{1}{3} = -\frac{1}{2}\left(\mathrm{ADF}(a) - \frac{1}{3}\right).$$

Since (c,d) is isoenergetic and $\mathrm{ADF}(c) = \mathrm{ADF}(d)$, Lemma 2.20 shows that $(f,g) = \sigma(c,d)$ is isoenergetic, that $\mathrm{ADF}(f) = \mathrm{ADF}(g) = \mathrm{ADF}(c)$, and that $\mathrm{CDF}(f,g) = \mathrm{CDF}(c,d)$. □

Now that we have seen how a single step of Construction 2.36 transforms the ADF and CDF of its input pair, we shall now investigate the effect of multiple steps.

Theorem 2.40. *Let (f,g) be an isoenergetic Golay pair with $\operatorname{ord} f = \operatorname{ord} g \neq \infty$, let m be an integer with $\max(\operatorname{len} f, \operatorname{len} g) \leq m$, and let $\gamma = (\gamma_0, \gamma_1, \ldots)$ be a sequence of transformations from* SGol. *Let $(f^{(n)}, g^{(n)}) = \mathrm{GRS}_\gamma^n((f,g), m)$ for each $n \in \mathbb{N}$. Then for each $n \in \mathbb{N}$, the pair $(f^{(n)}, g^{(n)})$ is an isoenergetic Golay pair with $\operatorname{ord}(f^{(n)}, g^{(n)}) = \operatorname{ord}(f,g) \neq \infty$, $\max(\operatorname{len} f^{(n)}, \operatorname{len} g^{(n)}) \leq 2^n m$, $\mathrm{ADF}(f^{(n)}) = \mathrm{ADF}(g^{(n)})$, and*

$$\mathrm{ADF}(f^{(n)}) - \frac{1}{3} = \left(-\frac{1}{2}\right)^n \left(\mathrm{ADF}(f) - \frac{1}{3}\right)$$
$$\mathrm{CDF}(f^{(n)}, g^{(n)}) - \frac{2}{3} = \left(-\frac{1}{2}\right)^n \left(\mathrm{CDF}(f,g) - \frac{2}{3}\right)$$

for each $n \in \mathbb{N}$. Thus

$$\lim_{n\to\infty} \mathrm{ADF}(f^{(n)}) = \lim_{n\to\infty} \mathrm{ADF}(g^{(n)}) = \frac{1}{3},$$
$$\lim_{n\to\infty} \mathrm{CDF}(f^{(n)}, g^{(n)}) = \frac{2}{3}.$$

PROOF. The asymptotic results follow immediately from the preceding formulae for the demerit factors, and the proof of all the non-asymptotic results proceeds by induction on n. If $n = 0$, then Lemma 2.9 shows that $\mathrm{ADF}(f) = \mathrm{ADF}(g)$, and then Lemmas 2.19 and 2.20 and Remark 2.22 show that $(f^{(0)}, g^{(0)}) = \gamma_0(f, g)$ is an isoenergetic Golay pair with $\mathrm{ord}(f^{(0)}, g^{(0)}) = \mathrm{ord}(f, g) \neq \infty$, $\max(\operatorname{len} f^{(0)}, \operatorname{len} g^{(0)}) = \max\{\operatorname{len} f, \operatorname{len} g\} \leq m$, $\mathrm{ADF}(f^{(0)}) = \mathrm{ADF}(g^{(0)}) = \mathrm{ADF}(f) = \mathrm{ADF}(g)$, and $\mathrm{CDF}(f^{(0)}, g^{(0)}) = \mathrm{CDF}(f, g)$. If $n > 0$ and we assume that our conclusions hold for $(f^{(n-1)}, g^{(n-1)})$, then Proposition 2.39 shows that our conclusions hold for $(f^{(n)}, g^{(n)})$. □

Remark 2.41. *Let m be a positive integer and let (f, g) be be a unimodular Golay pair of length m with $\operatorname{ord} f = \operatorname{ord} g$. Let $\gamma = (\gamma_0, \gamma_1, \ldots)$ be a sequence of transformations from* UGol. *Then (f, g), m, and γ satisfy the hypotheses of Theorem 2.40, and in addition to the conclusions therefrom, we also know that for each $n \in \mathbb{N}$ the pair $(f^{(n)}, g^{(n)})$ is unimodular and length $2^n m$. To see this, note that (f, g) is isoenergetic by Lemma 2.10, that f and g are of positive length and equal order, so $\operatorname{ord} f = \operatorname{ord} g \neq \infty$, and that* UGol *is a subgroup of* SGol*, so that (f, g), m, and γ satisfy all the hypotheses of Theorem 2.40, while Construction 2.36 shows that $(f^{(n)}, g^{(n)})$ is unimodular and of length $2^n m$ for every $n \in \mathbb{N}$.*

Remark 2.42. *Let m be a positive integer and let (f, g) be be a binary Golay pair of length m with $\operatorname{ord} f = \operatorname{ord} g$. Let $\gamma = (\gamma_0, \gamma_1, \ldots)$ be a sequence of transformations from* BGol. *Then (f, g), m, and γ satisfy the hypotheses of Theorem 2.40, and in addition to the conclusions therefrom, we also know that for each $n \in \mathbb{N}$ the pair $(f^{(n)}, g^{(n)})$ is binary and length $2^n m$. To see this, note that binary sequences are unimodular and* BGol *is a subgroup of* UGol*, so we may use Remark 2.41, and then Construction 2.36 shows that the pair $(f^{(n)}, g^{(n)})$ is binary for every $n \in \mathbb{N}$.*

2.8. An Iterated Golay Interleaving Construction

In this section, we shall explore how an iterated interleaving construction for Golay pairs influences demerit factors. In the previous section, we looked at the Golay-Rudin-Shapiro construction, which was noted to be a special case of Golay's concatenation construction. We present the analogous special case of Golay's interleaving (Construction 2.32), where we use 1 for the parameter m and $(1,1)$ for the first pair (a,b) in that construction.

Construction 2.43 (Simple Golay interleaving)**.** *If* $(a,b) \in \mathbb{C}[z,z^{-1}]^2$ *with* $\operatorname{ord} a + \deg a = \operatorname{ord} b + \deg b$*, then we set*

$$\begin{aligned}\operatorname{SGI}(a,b) &= \operatorname{scal}_{1,z^{2(\operatorname{ord} a+\deg a)}}\left((\operatorname{scal}_{1,z}(1,1)) \ltimes (\operatorname{subs}_{z^2}(a,b))\right)\\ &= \left(a(z^2) + zb(z^2), b^\ddagger(z^2) - za^\ddagger(z^2)\right).\end{aligned}$$

We prove some properties of our construction.

(i). If (a,b) *is a Golay pair, then so is* $\operatorname{SGI}(a,b)$*.*

(ii). If $\operatorname{len} a = \operatorname{len} b$*, then* $\operatorname{SGI}(a,b)$ *is a pair of order* $2\operatorname{ord} a$ *and length* $2\operatorname{len} a$*.*

(iii). If (a,b) *is a unimodular (resp., binary) pair with* $\operatorname{len} a = \operatorname{len} b$*, then* $\operatorname{SGI}(a,b)$ *is a unimodular (resp., binary) pair of length* $2\operatorname{len} a$*.*

Proof. The claim that $\operatorname{SGI}(a,b)$ is a Golay pair whenever (a,b) is follows from Lemma 2.19 and Corollary 2.29 and the fact that $(1,1)$ is a Golay pair. In the second and third claims, the additional assumption that $\operatorname{len} a = \operatorname{len} b$ along with the original assumption $\operatorname{ord} a + \deg a = \operatorname{ord} b + \deg b$ imply that $\operatorname{ord} a = \operatorname{ord} b$, and then the second expression for $\operatorname{SGI}(a,b)$ makes clear that $\operatorname{ord}\operatorname{SGI}(a,b) = 2\operatorname{ord} a$ and $\operatorname{len}\operatorname{SGI}(a,b) = 2\operatorname{len} a$ and that $\operatorname{SGI}(a,b)$ is unimodular (resp., binary) if $(a,b$ is unimodular (resp., binary). □

We now iterate the previous construction, also allowing for the application of transformations from SGol.

Construction 2.44 (Iterated simple Golay interleaving)**.** *Let* (a,b) *be a pair in* $\mathbb{C}[z,z^{-1}]^2$ *with* $\operatorname{len} a = \operatorname{len} b$ *and* $\operatorname{ord} a = \operatorname{ord} b$ *and let* $\gamma = (\gamma_0, \gamma_1, \ldots)$ *be a sequence of transformations from* SGol*.*

For $n \in \mathbb{N}$, we define $\mathrm{SGI}_\gamma^n(a,b)$ recursively by setting $\mathrm{SGI}_\gamma^0(a,b) = \gamma_0(a,b)$ and for $n > 0$ setting $\mathrm{SGI}_\gamma^{n+1} = \gamma_{n+1}(\mathrm{SGI}(\mathrm{SGI}_\gamma^n(a,b)))$. We prove some properties of our iterated construction.

(i). If (a,b) is a Golay pair, then so is $\mathrm{SGI}_\gamma^n(a,b)$ for every $n \in \mathbb{N}$.

(ii). $\mathrm{SGI}_\gamma^n(a,b)$ is a pair of length $2^n \operatorname{len} a$ and order $2^n \operatorname{ord} a$ for every $n \in \mathbb{N}$.

(iii). If (a,b) is a unimodular (resp., binary) pair and $\gamma_j \in \mathrm{UGol}$ (resp., $\gamma_j \in \mathrm{BGol}$) for every $j \in \mathbb{N}$, then $\mathrm{SGI}_\gamma^n(a,b)$ is a unimodular (resp., binary) pair of length $2^n \operatorname{len} a$ and order $2^n \operatorname{ord} a$ for every $n \in \mathbb{N}$.

Proof. Throughout this proof, it is useful to note that if two sequences, f and g, meet any two of the three conditions (I) $\operatorname{len} f = \operatorname{len} g$, (II) $\operatorname{ord} f = \operatorname{ord} g$, or (III) $\operatorname{ord} f + \deg f = \operatorname{ord} g + \deg g$, then they must also meet the third condition. This is clear because the pair $(f,g) = (0,0)$ meets all three conditions, a pair consisting of one zero sequence and one nonzero sequence meets neither (I) nor (II), and a pair (f,g) of two nonzero sequences has $\operatorname{len} f = \deg f - \operatorname{ord} f + 1$ and $\operatorname{len} g = \deg g - \operatorname{ord} g + 1$.

For pairs of sequences with matching lengths and orders, transformations from SGol preserve the length and order (see Remark 2.22) while SGI doubles the length and order (see Construction 2.43), so one can inductively establish the second claim for each $n \in \mathbb{N}$, which allows the recursive construction to carry on to the next step, since the input of SGI must be a pair (f,g) with $\operatorname{ord} f + \deg f = \operatorname{ord} g + \deg g$. The third claim follows from the second along with an induction using the final result in Construction 2.43 along with Remark 2.24 (for unimodular sequences) or Remark 2.26 (for binary sequences). The first claim is proved by induction using Lemma 2.19 and Construction 2.43. □

Example 2.45. *We consider how Construction 2.44 can be used to construct Golay pairs, using the same Boolean function formalism described in Example 2.37. If (a,b) is a binary Golay pair of length 2^n whose sequences correspond to Boolean functions $A(x_0,\ldots,x_{n-1})$ and $B(x_0,x_1,\ldots,x_{n-1})$, respectively, then it is not hard to show that $\mathrm{SGI}(a,b)$ is the binary Golay pair of length 2^{n+1} whose sequences correspond to the Boolean functions*

$$(1-x_0)A(x_1,\ldots,x_n) + x_0 B(x_1,\ldots,x_n)$$

and

$$(1-x_0)B(x_1+1,\ldots,x_n+1)+x_0A(x_1+1,\ldots,x_n+1)+x_0,$$

respectively. Therefore, if we start with (a,b) *equal to the binary Golay pair* $(1,1)$ *of length* 1*, and if we let* $\gamma=(\gamma_0,\gamma_1,\ldots)$ *where every* γ_j *is the identity transformation, then it is not hard to use induction to show that* $\mathrm{SGI}_\gamma^n(a,b)$ *is the binary Golay pair of length* 2^n *whose sequences correspond to the Boolean functions* 0 *and* 0 *(if* $n=0$*),* 0 *and* x_0 *(if* $n=1$*), or*

$$x_{n-2}x_{n-1}+\sum_{j=0}^{n-3}x_jx_{j+2}+\sum_{j=0}^{n-3}x_j \text{ and}$$
$$x_{n-2}x_{n-1}+\sum_{j=0}^{n-3}x_jx_{j+2}+\sum_{j=0}^{n-3}x_j+x_1+1$$

(for $n\geq 2$*), where we construe empty sums in these expressions as* 0 *when* $n=2$*.*

On the other hand, we can obtain very different sequences starting from the same initial pair $(1,1)$ *if we use some non-identity transformations between the stages of simple Golay interleaving. Recall from Example 2.37 that conjugate reversal of the binary sequence of length* 2^n *associated with Boolean function* $F(x_0,\ldots,x_{n-1})$ *changes it into the binary sequence of length* 2^n *associated with Boolean function* $F(x_0+1,\ldots,x_{n-1}+1)$*. Now we let* γ_j *be the identity map for all even* j*, but let* $\gamma_j=\mathrm{crev}$ *for all odd* j*. Then with some care one can prove by induction that if* (a,b) *is the binary Golay pair* $(1,1)$ *of length* 1*, then* $\mathrm{SGI}_\gamma^n(a,b)$ *is the binary Golay pair of length* 2^n *associated to the Boolean functions* 0 *and* 0 *(for* $n=0$*),* 0 *and* x_0 *(for* $n=1$*),* x_0x_1 *and* $x_0x_1+x_1+1$ *(for* $n=2$*), or* $C(x_0,\ldots,x_n)$ *and* $D(x_0,\ldots,x_n)$ *(for* $n\geq 3$*), where*

$$\begin{aligned}C(x_0,\ldots,x_n)=x_{n-3}x_{n-1}&+\sum_{j=0}^{\lfloor(n-2)/2\rfloor}x_{n-2-2j}x_{n-1-2j}\\&+\sum_{j=0}^{\lfloor(n-5)/2\rfloor}x_{n-5-2j}x_{n-2-2j}\\&+\sum_{j=0}^{n-4}x_j+x_{n-2}+(n+1)x_0+\binom{n-1}{2},\end{aligned}$$

and

$$D(x_0,\ldots,x_n)=C(x_0,\ldots,x_n)+nx_0+(n+1)x_2,$$

and we construe empty sums as 0 *when* n *is small.*

Now we investigate how a single step of Construction 2.44 affects demerit factors.

Proposition 2.46. *Let* (a, b) *be an isoenergetic Golay pair with* $\operatorname{len} a = \operatorname{len} b > 0$ *and* $\operatorname{ord} a = \operatorname{ord} b$. *Let* $\sigma \in \mathrm{SGol}$. *If* $(f, g) = \sigma(\mathrm{SGI}(a, b))$, *then* (f, g) *is an isoenergetic Golay pair with* $\operatorname{len}(f, g) = 2 \operatorname{len}(a, b) > 0$, $\operatorname{ord}(f, g) = 2 \operatorname{ord}(a, b)$, $\mathrm{ADF}(f) = \mathrm{ADF}(g)$, *and*

$$\mathrm{ADF}(f) - \frac{1}{3} = -\frac{1}{2}\left(\mathrm{ADF}(a) - \frac{1}{3}\right) + W(a, b)$$
$$\mathrm{CDF}(f, g) - \frac{2}{3} = -\frac{1}{2}\left(\mathrm{CDF}(a, b) - \frac{2}{3}\right) - W(a, b),$$

where

$$W(a, b) = \frac{2 \operatorname{Re}(\overline{z}(a\bar{b})^2)_0}{(|a|_0^2 + |b|_0^2)^2}.$$

Proof. Let $(c, d) = \mathrm{SGI}(a, b)$, so that $(f, g) = \sigma(c, d)$. From Construction 2.43 we know that (c, d) is a Golay pair with $\operatorname{len}(c, d) = 2 \operatorname{len}(a, b) > 0$ and $\operatorname{ord}(c, d) = 2 \operatorname{ord}(a, b)$. Then by Lemma 2.19 and Remark 2.22, we can see that $(f, g) = \sigma(c, d)$ is a Golay pair with $\operatorname{len}(f, g) = 2 \operatorname{len}(a, b) > 0$ and $\operatorname{ord}(f, g) = 2 \operatorname{ord}(a, b)$.

Let $(c', d') = (1, z) \ltimes (a(z^2), b(z^2))$ so that $(c, d) = \mathrm{scal}_{1, z^{2(\operatorname{ord} a + \deg a)}}(c', d')$. Because $|z|^2 = 1$, we see that $|c|^2$, $|d|^2$, and $|cd|^2$ are the same as $|c'|^2$, $|d'|^2$, and $|c'd'|^2$, respectively, and we can calculate these by using Lemmas 2.28 and 2.30 on (c', d') to obtain

$$|c|_0^2 = |a(z^2)|_0^2 + |b(z^2)|_0^2 + 2 \operatorname{Re}\left(\overline{z} a(z^2) \overline{b(z^2)}\right)_0$$
$$|d|_0^2 = |a(z^2)|_0^2 + |b(z^2)|_0^2 - 2 \operatorname{Re}\left(\overline{z} a(z^2) \overline{b(z^2)}\right)_0$$
$$|cd|_0^2 = |a(z^2)|_0^4 + |b(z^2)|_0^4 - 2 \operatorname{Re}\left(\left(\overline{z} a(z^2) \overline{b(z^2)}\right)^2\right)_0.$$

Notice that only monomials of odd degree occur in $\overline{z} a(z^2) \overline{b(z^2)}$, so there is no constant term in it or its conjugate. Thus the final term in each of the first two expressions vanishes. Further, we note

that $\mathrm{Re}((\overline{z}a(z^2)\overline{b(z^2)})^2)_0 = \mathrm{Re}(\overline{z}(a\overline{b})^2)_0$. As such,

$$|c|_0^2 = |a|_0^2 + |b|_0^2$$
$$|d|_0^2 = |a|_0^2 + |b|_0^2$$
$$|cd|_0^2 = |a|_0^4 + |b|_0^4 - 2\,\mathrm{Re}\left(\overline{z}(a\overline{b})^2\right)_0.$$

In view of (2.7), the first two equations show that (c, d) is isoenergetic, so Lemma 2.9 shows that $\mathrm{ADF}(c) = \mathrm{ADF}(d)$. Now we use (2.9) and the expressions above to see that

$$\mathrm{CDF}(c, d) = \frac{|a|_0^4 + |b|_0^4}{\left(|a|_0^2 + |b|_0^2\right)^2} - \frac{2\,\mathrm{Re}\left(\overline{z}(a\overline{b})^2\right)_0}{\left(|a|_0^2 + |b|_0^2\right)^2}.$$

Then use Lemma 2.38 and the definition of $W(a, b)$ in the statement of this lemma to see that

$$\mathrm{CDF}(c, d) = 1 - \frac{\mathrm{CDF}(a, b)}{2} - W(a, b),$$

and then we can subtract $2/3$ from both sides to obtain

$$\mathrm{CDF}(c, d) - \frac{2}{3} = -\frac{1}{2}\left(\mathrm{CDF}(a, b) - \frac{2}{3}\right) - W(a, b).$$

Since (c, d) is a Golay pair with $\mathrm{ADF}(c) = \mathrm{ADF}(d)$, Theorem 2.1 tells us that $\mathrm{CDF}(c, d)+\mathrm{ADF}(c) = 1$, and similarly $\mathrm{CDF}(a, b) + \mathrm{ADF}(a) = 1$ because $\mathrm{ADF}(a) = \mathrm{ADF}(b)$ by Lemma 2.9, so we can negate both sides of the last equation to obtain

$$\mathrm{ADF}(c) - \frac{1}{3} = -\frac{1}{2}\left(\mathrm{ADF}(a) - \frac{1}{3}\right) + W(a, b).$$

Since (c, d) is isoenergetic and $\mathrm{ADF}(c) = \mathrm{ADF}(d)$, Lemma 2.20 shows that $(f, g) = \sigma(c, d)$ is isoenergetic, that $\mathrm{ADF}(f) = \mathrm{ADF}(g) = \mathrm{ADF}(c)$, and that $\mathrm{CDF}(f, g) = \mathrm{CDF}(c, d)$. □

We now wish to investigate the autocorrelation and crosscorrelation demerit factors of $\mathrm{SGI}_\gamma^n(a, b)$ from Construction 2.44 by applying the last proposition multiple times, but the term $W(a, b)$ that appears in that proposition can make calculations troublesome. We find that if we restrict σ to come from a carefully selected subgroup of SGol, then when we use the output pair (f, g) from the proposition as an input into the same proposition (for the next step of Construction 2.44), we will

find that $W(f,g) = 0$, thus simplifying the calculation. We first describe the subgroup of SGol that allows for this simplification.

Definition 2.47 (Restricted Golay Group RGol)**.** *The restricted Golay group, written* RGol*, is the subgroup of* SGol *generated by* swap*,* $\mathrm{crev} \circ \mathrm{crev}'$*,* srev*, all transformations* $\mathrm{scal}_{u,v}$ *where* u, v *are nonzero complex numbers with* $|u| = |v|$*, and all transformations* subs_{wz}*, where* w *is a unimodular complex number.*

Remark 2.48. *The only difference between the generating set of* RGol *and that of* SGol *is that* RGol *has the single generator* $\mathrm{crev} \circ \mathrm{crev}'$ *in place of the two generators* crev *and* crev' *for* SGol*. Thus* RGol *is a subgroup of* SGol*, and indeed a proper subgroup because it is straightforward to show that if* $(a,b) = (1+z+z^2-z^3, 1+z-z^2+z^3)$ *and* $(f,g) = \gamma(a,b)$ *for some* $\gamma \in$ RGol*, then* fg *has terms of both even and odd degree, but* $a^\ddagger b$ *has only terms of even degree, so* $\gamma(a,b) = (f,g) \neq (a^\dagger, b) = \mathrm{crev}(a,b)$*, and so* $\mathrm{crev} \in \mathrm{SGol} \setminus \mathrm{RGol}$*.*

Definition 2.49 (Restricted Unimodular Golay Group RUGol)**.** *The restricted unimodular Golay group, written* RUGol*, is* $\mathrm{RGol} \cap \mathrm{UGol}$*.*

Remark 2.50. *Note that* RUGol *contains* swap*,* $\mathrm{crev} \circ \mathrm{crev}'$*,* srev*, all transformations* $\mathrm{scal}_{u,v}$ *where* u, v *are unimodular complex numbers, and all transformations* subs_{wz}*, where* w *is a unimodular complex number, because all these transformations are in both* RGol *and* UGol*.*

Definition 2.51 (Restricted Binary Golay Group RBGol)**.** *The restricted binary Golay group, written* RBGol*, is* $\mathrm{RGol} \cap \mathrm{BGol}$*.*

Remark 2.52. *Note that* RBGol *contains* swap*,* $\mathrm{crev} \circ \mathrm{crev}'$*,* srev*,* $\mathrm{scal}_{-1,1}$*,* $\mathrm{scal}_{1,-1}$*, and* subs_{-z}*, because all these transformations are in both* RGol *and* BGol*. Recall that* $\mathrm{crev} \circ \mathrm{crev}'$ *and* srev *have the same effect when applied to sequences with real terms.*

The following lemma shows that if we use a transformation σ from RGol in Proposition 2.46, then the output pair (f,g) from that proposition has the property that $W(f,g) = 0$ when we use (f,g) again as an input pair for that proposition.

Lemma 2.53. *Let* $(a, b) \in \mathbb{C}[z, z^{-1}]^2$ *with* $\operatorname{ord} a + \deg a = \operatorname{ord} b + \deg b$*, let* $\sigma \in \mathrm{RGol}$*, and let* $(f, g) = \sigma(\mathrm{SGI}(a, b))$*. Then* $\mathrm{Re}(\overline{z}(f\overline{g})^2)_0 = 0$.

Proof. Let $(c, d) = \mathrm{SGI}(a, b)$, so that $(f, g) = \sigma(c, d)$. From the formula for $\mathrm{SGI}(a, b)$ in Construction 2.43, and using the hypothesis that $\operatorname{ord} a + \deg a = \operatorname{ord} b + \deg b$, we see that

$$c\overline{d} = \left(a(z^2) + zb(z^2)\right)\left(z^{-2(\operatorname{ord} b+\deg b)}b(z^2) - z^{-2(\operatorname{ord} a+\deg a)-1}a(z^2)\right),$$

so that

$$c\overline{d} = -z^{-2(\operatorname{ord} a+\deg a)-1}(a(z^2)^2 - z^2 b(z^2)^2),$$

which has only odd degree terms. We say that a Laurent polynomial $x(z)$ is *parity-pure* to mean that $x(z)$ does not have both a term of even degree and a term of odd degree, and we say that a pair (x, y) of Laurent polynomials is *parity-pure* to mean that the Laurent polynomial $x\overline{y}$ is parity-pure. So (c, d) is parity-pure. We claim that $(f, g) = \sigma(c, d)$ is also parity-pure. Recall the generating set of RGol specified in Definition 2.47. Since this set of generators is closed under inversion (see Lemma 2.13, and note that $\mathrm{crev} \circ \mathrm{crev}'$ is an involution), σ is a composition of some of these generators. We claim that any such generator η, when applied to a parity-pure pair $(x, y) \in \mathbb{C}[z, z^{-1}]^2$, produces another parity-pure pair $(x', y') = \eta(x, y)$, because of the following considerations.

- If $\eta = \mathrm{swap}$, then $x'\overline{y'} = y\overline{x} = \overline{x\overline{y}}$, which is parity-pure.
- If $\eta = \mathrm{crev} \circ \mathrm{crev}'$, then $x'\overline{y'} = x^{\ddagger}\overline{y^{\ddagger}} = z^{\operatorname{ord} x+\deg x-\operatorname{ord} y-\deg y}\overline{x\overline{y}}$, which is parity-pure.
- If $\eta = \mathrm{srev}$, then

$$\begin{aligned} x'\overline{y'} &= z^{\operatorname{ord} x+\deg x-\operatorname{ord} y-\deg y}x(z^{-1})\overline{y(z^{-1})} \\ &= z^{\operatorname{ord} x+\deg x-\operatorname{ord} y-\deg y}(x\overline{y})(z^{-1}), \end{aligned}$$

  which is parity-pure.
- If $\eta = \mathrm{scal}_{u,v}$ where u and v are nonzero complex numbers, then $x'\overline{y'} = u\overline{v}x\overline{y}$, which is parity-pure.
- If $\eta = \mathrm{subs}_{wz}$ where w is a unimodular complex number, then $x'\overline{y'} = x(wz)\overline{y(wz)}$. But if we let $h = \overline{y}$, then $\overline{y(wz)} = h(\overline{w}^{-1}z) = h(wz)$, so that $x'\overline{y'} = (x\overline{y})(wz)$, which is parity-pure.

Thus we conclude that $(f, g) = \sigma(c, d)$ is parity-pure, and so $\overline{z}(f\overline{g})^2$ has only terms of odd degree, so that $\operatorname{Re}(\overline{z}(f\overline{g})^2)_0 = 0$. □

Now we can use Proposition 2.46 with Lemma 2.53 to analyze demerit factors of pairs produced by Construction 2.44.

Theorem 2.54. *Let (f, g) be an isoenergetic Golay pair with $\operatorname{len} f = \operatorname{len} g > 0$ and $\operatorname{ord} f = \operatorname{ord} g$. Let $\gamma = (\gamma_0, \gamma_1, \ldots)$ be a sequence of transformations from* RGol. *Let $(f^{(n)}, g^{(n)}) = \operatorname{SGI}_\gamma^n(f, g)$ for each $n \in \mathbb{N}$. Then for each $n \in \mathbb{N}$, the pair $(f^{(n)}, g^{(n)})$ is an isoenergetic Golay pair with $\operatorname{len}(f^{(n)}, g^{(n)}) = 2^n \operatorname{len}(f, g) > 0$, $\operatorname{ord}(f^{(n)}, g^{(n)}) = 2^n \operatorname{ord}(f, g)$, and $\operatorname{ADF}(f^{(n)}) = \operatorname{ADF}(g^{(n)})$. We have $\operatorname{ADF}(f^{(0)}) = \operatorname{ADF}(f)$ and $\operatorname{CDF}(f^{(0)}, g^{(0)}) = \operatorname{CDF}(f, g)$, and for $n > 0$ we have*

$$\operatorname{ADF}(f^{(n)}) - \frac{1}{3} = \left(-\frac{1}{2}\right)^n \left(\operatorname{ADF}(f) - \frac{1}{3}\right) + \left(-\frac{1}{2}\right)^{n-1} W_0$$

and

$$\operatorname{CDF}(f^{(n)}, g^{(n)}) - \frac{2}{3} = \left(-\frac{1}{2}\right)^n \left(\operatorname{CDF}(f, g) - \frac{2}{3}\right) - \left(-\frac{1}{2}\right)^{n-1} W_0,$$

where

$$W_0 = \frac{2 \operatorname{Re}((\overline{z}(f^{(0)}\overline{g^{(0)}})^2)_0}{(|f^{(0)}|_0^2 + |g^{(0)}|_0^2)^2}.$$

Thus

$$\lim_{n\to\infty} \operatorname{ADF}(f^{(n)}) = \lim_{n\to\infty} \operatorname{ADF}(g^{(n)}) = \frac{1}{3},$$
$$\lim_{n\to\infty} \operatorname{CDF}(f^{(n)}, g^{(n)}) = \frac{2}{3}.$$

Proof. The asymptotic results follow immediately from the formulae for the demerit factors, and the proof of all the non-asymptotic results proceed by induction on n. If $n = 0$, then Lemma 2.9 shows that $\operatorname{ADF}(f) = \operatorname{ADF}(g)$, and then Lemma 2.19, Lemma 2.20, and Remark 2.22 show that $(f^{(0)}, g^{(0)})$ is an isoenergetic Golay pair with $\operatorname{len}(f^{(0)}, g^{(0)}) = \operatorname{len}(f, g) > 0$, $\operatorname{ord}(f^{(0)}, g^{(0)}) = \operatorname{ord}(f, g)$, $\operatorname{ADF}(f^{(0)}) = \operatorname{ADF}(g^{(0)}) = \operatorname{ADF}(f) = \operatorname{ADF}(g)$, and $\operatorname{CDF}(f^{(0)}, g^{(0)}) = \operatorname{CDF}(f, g)$. For $n = 1$, Proposition 2.46 gives us all the desired results. If $n > 1$ and we assume the results hold for $(f^{(n-1)}, g^{(n-1)})$, then we apply Proposition 2.46, which immediately tells us that $(f^{(n)}, g^{(n)})$ is an

isoenergetic Golay pair of length $2^n \operatorname{len}(f,g)$ and order $2^n \operatorname{ord}(f,g)$ with $\operatorname{ADF}(f^{(n)}) = \operatorname{ADF}(g^{(n)})$. Furthermore, since Lemma 2.53 tells us that $W(f^{(n-1)}, g^{(n-1)}) = 0$, the demerit factor formulae that Proposition 2.46 supplies become

$$\operatorname{ADF}(f^{(n)}) - \frac{1}{3} = -\frac{1}{2}\left(\operatorname{ADF}(f^{(n-1)}) - \frac{1}{3}\right)$$
$$\operatorname{CDF}(f^{(n)}, g^{(n)}) - \frac{2}{3} = -\frac{1}{2}\left(\operatorname{CDF}(f^{(n-1)}, g^{(n-1)}) - \frac{2}{3}\right),$$

into which we substitute the values of $\operatorname{ADF}(f^{(n-1)})$ and $\operatorname{CDF}(f^{(n-1)}, g^{(n-1)})$ from the induction hypothesis to obtain the desired result. □

Remark 2.55. *Let (f,g) be a unimodular Golay pair of nonzero sequences with $\operatorname{ord} f = \operatorname{ord} g$. Let $\gamma = (\gamma_0, \gamma_1, \ldots)$ be a sequence of transformations from* RUGol*. Then (f,g) and γ satisfy the hypotheses of Theorem 2.54, and in addition to the conclusions therefrom, we also know that for each $n \in \mathbb{N}$ the pair $(f^{(n)}, g^{(n)})$ is unimodular. To see this, note that Lemma 2.10 shows that (f,g) is isoenergetic and $\operatorname{len} f = \operatorname{len} g$ (which is nonzero since the sequences are nonzero), and note that* RUGol *is a subgroup of* RGol*, so that (f,g) and γ satisfy all the hypotheses of Theorem 2.54, while Construction 2.44 shows that $(f^{(n)}, g^{(n)})$ is unimodular for every $n \in \mathbb{N}$.*

Remark 2.56. *Let (f,g) be a binary Golay pair of nonzero sequences with $\operatorname{ord} f = \operatorname{ord} g$. Let $\gamma = (\gamma_0, \gamma_1, \ldots)$ be a sequence of transformations from* RBGol*. Then (f,g) and γ satisfy the hypotheses of Theorem 2.54, and in addition to the conclusions therefrom, we also know that for each $n \in \mathbb{N}$ the pair $(f^{(n)}, g^{(n)})$ is binary. To see this, note that binary sequences are unimodular and* RBGol *is a subgroup of* RUGol*, so we may use Remark 2.55, and then Construction 2.44 shows that the pair $(f^{(n)}, g^{(n)})$ is binary for every $n \in \mathbb{N}$.*

2.9. Open Problems

In Sections 2.7 and 2.8, we observed that we can find infinitely many families of unimodular (or binary) Golay pairs whose sequences have autocorrelation demerit factors and crosscorrelation demerit factors that tend to 1/3 and 2/3, respectively, as the length of the sequences tends to infinity. So far, we have been unable to find a construction that creates a family of Golay pairs that deviates from these asymptotic demerit factor values. This leaves us with two open problems about the

existence of such a families of Golay pairs. Recall from Lemma 2.10 that if $p = (f, g)$ is a unimodular Golay pair consisting of nonzero sequences, then $\operatorname{len} f = \operatorname{len} g$ and $\operatorname{ADF}(f) = \operatorname{ADF}(g)$, so we can write $\operatorname{len} p$ and $\operatorname{ADF}(p)$ for the common values.

Open Problem 2.57. *Let $\{p_n\}_{n=1}^{\infty}$ be a sequence of binary Golay pairs with nonzero sequences such that $\operatorname{len} p_n \to \infty$ as $n \to \infty$. Must it be true that $\lim_{n\to\infty} \operatorname{ADF}(p_n) = 1/3$ and $\lim_{n\to\infty} \operatorname{CDF}(p_n) = 2/3$?*

Similarly, there is the open problem for the more general unimodular case.

Open Problem 2.58. *Let $\{p_n\}_{n=1}^{\infty}$ be a sequence of unimodular Golay pairs with nonzero sequences such that $\operatorname{len} p_n \to \infty$ as $n \to \infty$. Must it be true that $\lim_{n\to\infty} \operatorname{ADF}(p_n) = 1/3$ and $\lim_{n\to\infty} \operatorname{CDF}(p_n) = 2/3$?*

CHAPTER 3

Using Noise to Probe Recurrent Neural Network Structure and Prune Synapses

Published in *Advances in Neural Information Processing Systems 33* (NeurIPS, December 2020).

Edited for this dissertation.

Joint work with:

Rishidev Chaudhuri

Center for Neuroscience

Department of Mathematics

Department of Neurobiology, Physiology and Behavior

University of California, Davis

Davis, CA 95616

rchaudhuri@ucdavis.edu

3.1. Abstract

Many networks in the brain have sparse connectivity, and the brain prunes synapses during development and learning. With the complex inter-connectivity the brain possesses, determining which synapses it can safely prune is a difficult problem dependent on higher-order information than direct synaptic weights alone. How does the brain determine which synapses are redundant, and which are structurally vital? Noise is pervasive in neural systems, often considered an irritant to overcome. In this chapter, our results suggest that noise could contribute to synaptic pruning mechanisms, allowing the brain to probe its own network structure. We construct an anti-Hebbian, stochastic, local, unsupervised plasticity rule that either strengthens or prunes synapses using only synaptic weight and the noise-driven covariance of the neighboring neurons. For diagonally-dominant and symmetric linear and rectified-linear recurrent networks, we prove that this rule preserves the

spectrum of the original dense connectivity matrix, even when the fraction of pruned synapses asymptotically approaches 1. Finally, the plasticity rule is biologically-plausible and suggests a probative role for noise in neural computation.

3.2. Introduction

Pruning, or the elimination of synaptic connections, was first experimentally observed in the brain almost five decades ago, notably occurring dramatically during puberty and moderately in old age [**38**, **42**, **65**, **86**]. Not only is pruning known to occur within the brain, but it is also studied outside of the biological context in order to create sparsely connected artificial networks able to perform computation more efficiently than their dense counterparts, while simultaneously consuming less memory [**10**, **71**]. In addition, synaptic density is disrupted across a variety of diseases such as autism and schizophrenia, leading clinical investigators to hypothesize that atypical synaptic sparsity levels may be related to the development of some mental disorders [**64**, **66**, **94**, **96**]. If we can understand how the brain determines and maintains its sparse network structure, perhaps we can explain the experimentally observed changes in connection density through aging and disease. Furthermore, if we can bridge the gap between biological and artificial pruning algorithms, we may be able to replicate the remarkable energy efficiency the brain has in artificial neural networks, and ideally discover more about the brain by studying these artificial networks.

Unlike the hierarchical feed-forward nature of most artificial networks, highly-recurrent networks such as those in the brain have multiple pathways of information flow. This makes it difficult to classify which synapses are redundant and thus able to be safely pruned, and which are crucial to the function of the network. For example, even if a synapse between two neurons is strong, it may be the case that information can travel between these neurons through alternate pathways, rendering the aforementioned synapse redundant and safe to discard. A biologically-plausible pruning rule must determine this higher-order structure using only information locally available at the synapse.

In a highly-recurrent network with multiple pathways of information flow, it is difficult to determine which synapses are redundant and can be safely pruned, and which are important and should be retained. For example, even if a synapse between two neurons is strong, if information can travel between the neurons by alternative pathways then the synapse is redundant and can be

removed. A biologically-plausible pruning rule must determine this higher-order structure using information locally available at the synapse, both in space and time [**5**].

As far as the literature currently stands, neural systems seem noisy at multiple levels: neural activity contains large background fluctuations, responses to the same stimulus can be quite variable, and synapses often fail to propagate a signal [**16**, **21**, **48**, **81**]. In this chapter, we postulate that noise could play a vital computational role in synaptic pruning. Specifically, the pattern of activity correlations in a noise-driven network reflects higher-order network structure in precisely the manner required to prune redundant synapses (as predicted by a theoretical argument). We construct a local plasticity rule that either strengthens or prunes synapses with a probability given by the synaptic weight and the noise-driven variance and covariance of the two neighboring neurons. The plasticity rule is unsupervised and task-agnostic, and manages to yield a sparse network that preserves existing network dynamics, whatever they are. Thus, it could act alongside learning or during separate background pruning epochs (e.g., sleep).

We prove that, for diagonally-dominant, undirected linear and rectified linear networks, the pruning rule preserves multiple useful properties of the original network (including the spectrum and steady-state firing-rate variances), even when the fraction of removed synapses approaches 1. The theoretical results link neural network pruning and noise-driven dynamical systems to a powerful body of results in sampling-based graph sparsification [**6**, **83**, **84**] and to random matrix concentration of measure techniques [**1**, **73**, **74**, **90**].

3.3. Problem Setup

In this chapter, we will primarily consider linear neural networks of the form

$$\frac{d\boldsymbol{x}}{dt} = -D\boldsymbol{x} + W\boldsymbol{x} + \boldsymbol{b}(t) = Ax + \boldsymbol{b}(t) \tag{3.1}$$

The vector $\boldsymbol{x}$ represents the firing rate of N neurons, with each x_i the firing rate of the i-th neuron. The vector-valued function $\boldsymbol{b}(t)$ is the external input to the neurons (including biases), which we allow to vary over time. W is the connectivity matrix of synaptic weights between the neurons, with W_{ij} the connection strength from the j-th to the i-th neuron. D is a diagonal matrix representing the

intrinsic leak of activity (or the excitability of the neuron). Finally we define the matrix $A = -D+W$. We discuss generalizations to rectified linear networks in Section 3.7.

The pruning rule we describe in the following section manages to generate a sparse network with corresponding matrix A^{sp} with two properties. First, the number of edges in the pruned network (i.e., number of non-zero entries in A^{sp}) will be $O(N \log N)$. Indeed, this is dramatically fewer connections than the $O(N^2)$ possible edges in the original network. Second, the dynamics of the pruned network

$$\frac{d\boldsymbol{x}}{dt} = A^{sp}\boldsymbol{x} + \boldsymbol{b}(t) \tag{3.2}$$

will be similar to the dynamics of the original network in Eq. 3.1 (as demonstrated in Fig. 2).

To measure the similarity of A and A^{sp}, we adopt the notion of spectral similarity **[83, 84]** from the field of graph sparsification. That is, for any degree of error $\epsilon > 0$, we want to generate an A^{sp} that satisfies

$$|\boldsymbol{x}^T(A^{sp} - A)\boldsymbol{x}| \leq \epsilon|\boldsymbol{x}^T A\boldsymbol{x}| \quad \forall \boldsymbol{x} \in \mathbb{R}^N. \tag{3.3}$$

This notion of similarity is stronger than one might expect. For symmetric matrices, it requires that the eigenvalues of A^{sp} approximate the eigenvalues of A (and hence all the timescales of the resulting dynamics) to within a multiplicative factor of $1+\epsilon$ (see Appendix Section 3.9.1.3). It also manages to approximately preserve the orientation of the corresponding eigenspaces (again, see Appendix Section 3.9.1.3). This closeness is much stronger than low rank approximation (e.g., Principal Component Analysis), which only preserves the largest eigenvalues, or the fastest timescales. The timescales and activity patterns of the dynamical system in Eq. 3.1 are determined by the spectrum of A, and thus spectrum-preserving sparsification will approximately preserve dynamics as well.

3.4. An Unsupervised Noise-Driven Anti-Hebbian Pruning Rule

Let us consider the network in Eq. 3.1 when driven by independent noise at each node. We set $\boldsymbol{b}(t) = \boldsymbol{b} + \sigma\boldsymbol{\xi}(t)$ where $\boldsymbol{b}$ is an arbitrary vector of constant background input to the network, $\boldsymbol{\xi}(t)$ is a vector-valued function of IID unit variance Gaussian white noise at each point in time, and σ is the standard deviation of the input noise. Let $C = \mathbb{E}_t[\boldsymbol{x}\boldsymbol{x}^T]$ be the covariance matrix of the firing

rates in response to this white noise input (and note that this expectation is taken over time). For the synapse from neuron j to neuron i with weight w_{ij}, we define the sampling probability

$$p_{ij} = \begin{cases} Kw_{ij}\left(C_{ii} + C_{jj} - 2C_{ij}\right) & \text{for } w_{ij} > 0 \quad \text{(i.e., an excitatory synapse)} \\ K|w_{ij}|\left(C_{ii} + C_{jj} + 2C_{ij}\right) & \text{for } w_{ij} < 0 \quad \text{(i.e., an inhibitory synapse).} \end{cases} \tag{3.4}$$

Here C_{ii} and C_{jj} are the variances over time of the ith and jth neurons, and C_{ij} is their covariance over time. K is a proportionality constant that determines the density of the pruned network, which will have $NK/2$ total connections on average and thus average degree of $K/2$ per neuron (for unit variance noise and symmetric networks). Indeed, K will depend on the degree of approximation we desire, ϵ.

Now consider a pruning process that independently preserves or discards each edge with probability p_{ij} or $(1 - p_{ij})$, respectively, yielding A^{sp}. That is, for $i \neq j$,

$$A^{sp}_{ij} = \begin{cases} A_{ij}/p_{ij} & \text{with probability } p_{ij} \\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

It is worth noting that preserved edges are strengthened in magnitude, thanks to dividing by $p_{ij} < 1$. For the diagonal terms (i.e., leak / excitability) A^{sp}_{ii}, we either preserve the original diagonal and set $A^{sp}_{ii} = A_{ii}$, or define the perturbation $\Delta_i = \sum_{j \neq i} |A^{sp}_{ij}| - \sum_{j \neq i} |A_{ij}|$ (which is the change in total input to neuron i after pruning) and set $A^{sp}_{ii} = A_{ii} - \Delta_i$. Δ_i has zero mean and is likely to be small, so both of these processes will yield comparable diagonal elements. The interest in Δ_i is due to it biologically corresponding to changing the excitability of neuron i in response to a change in total input, and excitability is known to be homeostatically regulated in the brain **[91]**. We will refer to these as the "original diagonal" and "matched diagonal" settings respectively. The proofs apply to the "matched diagonal" setting. However, empirically, we observe similar results in the "original diagonal" regime, so we include it for its algorithmic simplicity. We will also refer to the pruning rule defined by Eqs. 3.4, 3.5 (in both diagonal settings) as **noise-prune** going forward.

The noise-prune rule is derived from a theoretical argument in the next section of this chapter, but it has an appealingly simple interpretation aside from the nuance of its proof. We now provide some intuition for why it might work. First, note that the probability to preserve a synapse depends

Algorithm 1 noise-prune, original diagonal setting

Input: $A \in \mathbb{R}^{N\times N}$ and error tolerance $\epsilon > 0$.
1. **For each** index (i, j) with $i \neq j$ set p_{ij} as in Eq. 3.4.
2. **Define** the sparse random matrix A^{sp} with off-diagonal entries

$$A^{sp}_{ij} = \begin{cases} \frac{A_{ij}}{p_{ij}} & \text{with probability } p_{ij} \\ 0 & \text{with probability } 1 - p_{ij}. \end{cases},$$

and diagonal entries $A^{sp}_{ii} = A_{ii}$.
Output: $A^{sp} \in \mathbb{R}^{N\times N}$.

Algorithm 2 noise-prune, matched diagonal setting

Input: $A \in \mathbb{R}^{N\times N}$ and error tolerance $\epsilon > 0$.
1. **For each** index (i, j) with $i \neq j$ set p_{ij} as in Eq. 3.4.
2. **Set** $\Delta_i = \sum_{j\neq i} |A^{sp}_{ij}| - \sum_{j\neq i} |A_{ij}|$.
3. **Define** the sparse random matrix A^{sp} with off-diagonal entries

$$A^{sp}_{ij} = \begin{cases} \frac{A_{ij}}{p_{ij}} & \text{with probability } p_{ij} \\ 0 & \text{with probability } 1 - p_{ij}. \end{cases},$$

and diagonal entries $A^{sp}_{ii} = A_{ii} - \Delta_i$.
Output: $A^{sp} \in \mathbb{R}^{N\times N}$.
Note: If A is symmetric and diagonally dominant, we are guaranteed

$$|\boldsymbol{x}^T(A^{sp} - A)\boldsymbol{x}| \leq \epsilon|\boldsymbol{x}^T A\boldsymbol{x}| \quad \forall \boldsymbol{x} \in \mathbb{R}^N.$$

on the magnitude of its weight, $|w_{ij}|$. Thus, holding all else equal, synapses with larger weight are more important and thus more likely to be preserved. The other factor in the sampling probability is $(C_{ii} + C_{jj} \pm 2C_{ij}) = 2\tilde{C}_{ij}(1 \pm C_{ij}/\tilde{C}_{ij})$, where $\tilde{C}_{ij} = (C_{ii} + C_{jj})/2$ is the mean variance of nodes i and j. We will refer to this factor as the combination-of-covariances (or "comb-cov") term. With the comb-cov term written as above, it becomes clear that preservation probability is proportional to $\tilde{C}_{ij}$, indicating that nodes with higher variance are considered more important. As such, their connections are more likely to be preserved (as in a PCA-like approximation). Finally, there is an anti-Hebbian term that, for excitatory synapses, takes the form $(1 - C_{ij}/\tilde{C}_{ij})$. Synapses are thus likely to be preserved if they are weakly or anti-correlated despite having an excitatory connection. The equivalent term for inhibitory synapses is $(1 + C_{ij}/\tilde{C}_{ij})$. The sign of the covariance is flipped, reflecting that inhibitory connections are expected to have anti-correlated neurons.

The covariance of neurons i and j depends both on the magnitude of the direct connection between them (i.e., $|w_{ij}|$) and on indirect connections through the rest of the network. Neurons

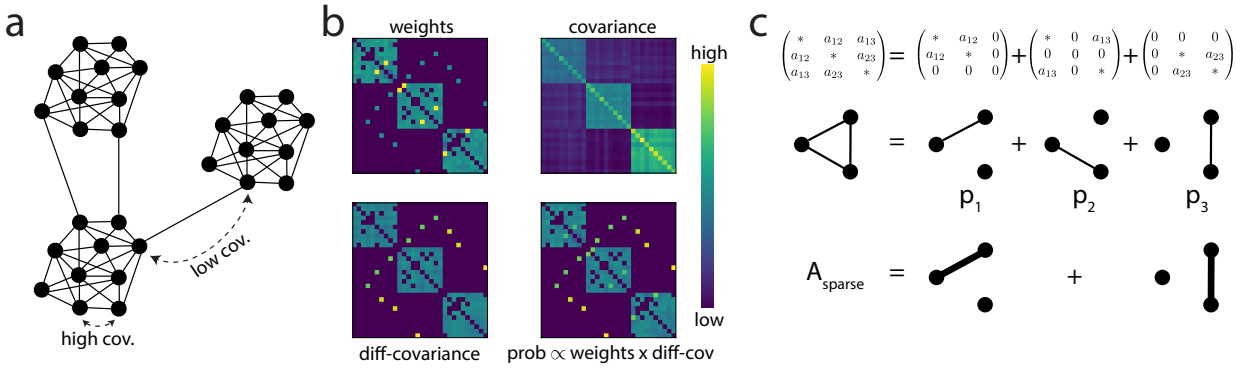

Figure 1: **A noise-driven unsupervised synaptic pruning rule.** (a) Schematic of a network where noisy fluctuations reflect higher-order connectivity structure. Network has 3 densely-connected clusters, with a few long-range connections. Covariance between neurons within a cluster is high compared to neurons participating in different clusters. (b) Pruning rule uses weights and covariances to identify important synapses. Top left: connection weights in a network with 3 densely-connected clusters and sparse connections between clusters. Also note the presence of a few strong connections within each cluster. Top right: covariance when driven by noise. Bottom left: difference of covariances (as in Eq. 3.4). Bottom right: sampling probabilities from pruning rule. The rule correctly identifies that the sparse connections between clusters are important and assigns them higher probability, along with the handful of exceptionally strong connections within a cluster. Most connections within a cluster are redundant and given lower probability. (c) Schematic of proof strategy. The original network (shown as a matrix in the top row and as a graph in the middle row) can be written as a sum of edge pieces. The edges are assigned sampling probabilities (p_1, p_2, p_3) that depend on weight and covariance. A given application of noise-prune yields a sparse network (bottom row) where some connections are preserved and strengthened (first and third edges) and others are pruned (second edge). For appropriate probabilities, the spectrum of A^{sp} is close to that of the original network.

that are highly correlated are likely to have multiple indirect connections, suggesting that the direct connection between them is redundant and can be pruned (see graphics in Fig. 1a,b). In a sense, the pruning rule can be understood as probing whether neurons are more correlated than expected given the weight of their direct connection. If they are, the connection is likely to be redundant, and is thus likely to be pruned by noise-prune.

3.5. Proofs

We derive this pruning rule from a two-part theoretical argument, largely inspired by Spielman & Srivastava (2011), with slight extension to signed symmetric diagonally-dominant neural network matrices (rather than Graph Laplacians, which their work was focused on). We begin by considering a sampling-based approach to pruning that independently strengthens or removes each synapse of

a network with some sampling probability (as in Eq. 3.5) and show that these probabilities that preserve network dynamics. Following this, we show that these theoretically-derived probabilities can be simply written in terms of the covariance of the network activity when driven by noisy fluctuations. This representation of sampling probabilities in terms of covariance is novel to this study, and perhaps of special note, as they make the pruning algorithm local in nature. Thus, there exists a simple and biologically-plausible way for neural networks to compute the sampling probabilities using only local information available to the synapse.

We note that the proof we present requires the matrix A to be symmetric (corresponding to an undirected graph) and diagonally-dominant (corresponding to highly leaky neurons), but the noise-prune rule itself need not have the same theoretical restrictions. These are strong restrictions that do not typically apply to neural networks, and we discuss generalizations and limitations later, including preliminary empirical results that show that noise-prune can perform quite well on networks with asymmetric, general diagonal matrices (see Fig. 2).

3.5.1. Derivation of Probabilities. Assume that the connectivity matrix A from Eq. 3.1 is both symmetric ($A_{ij} = A_{ji}$) and diagonally-dominant ($|A_{ii}| \geq \sum_{j \neq i} |A_{ij}| = \sum_{j \neq i} |w_{ij}|$). The diagonal entries of A are negative, reflecting the inherent leakiness of the neurons, and thus A is negative definite. We pause to note that the eigenvalues of A must be negative for the linear system to be stable (which is generally desirable in neuroscience to avoid non-biological neural activity), but the argument can be extended to non-invertible matrices by restricting to the subspace of nonzero eigenvalues, i.e. the range of A **[83]**). For notational convenience of working with positive definite matrices rather than negative definite ones, we define the positive definite matrix $B = -A$ and consider B instead of A in this section.

Given an index (i, j), $i > j$, with weight w_{ij}, define the edge matrix $X^{(i,j)}$ to have ith and jth diagonal entries $X^{(i,j)}_{ii} = X^{(i,j)}_{jj} = |w_{ij}|$. Set the (i, j)th and (j, i)th off-diagonal entries $X^{(i,j)}_{ij} = X^{(i,j)}_{ji} = -w_{ij}$ and remaining entries 0 (Fig. 1c for a schematic). That is, $X^{(i,j)}$ has off-diagonal entries equal to negative edge weight and diagonal entries equal to edge weight magnitude. Next, for diagonal indices (i, i), define the single-entry diagonal matrix $X^{(i,i)}_{ii} = B_{ii} - \sum_{j \neq i} |w_{ij}|$, with remaining entries 0. Since B is diagonally-dominant with positive diagonal, the single non-zero entry of $X^{(i,i)}$ is positive. For simplicity, we consider matrices where $X^{(i,i)} = 0$ (i.e., $B_{ii} = \sum_{j \neq i} |w_{ij}|$, so

that the leakiness of neurons is equal to their input weights), but it is straightforward to include non-zero $X^{(i,i)}$ where the leak is stronger (Appendix Section 3.9.1.1). Using these newly defined $X^{(i,j)}$, B can be written as a sum over edge matrices as $B = \sum_{i>j} X^{(i,j)}$.

Now define the random matrix $\tilde{X}^{ij}$ with a single Bernoulli entry as

$$\tilde{X}^{ij} = \begin{cases} X^{(i,j)}/p_{ij} & \text{with probability } p_{ij} \\ 0 & \text{with probability } 1 - p_{ij}, \end{cases} \tag{3.6}$$

and define $B^{sp} = \sum_{i>j} \tilde{X}^{ij}$. For any choice of p_{ij}, $\mathbb{E}[B^{sp}] = B$ because

$$\mathbb{E}[B^{sp}] = \sum_{i>j} \mathbb{E}[\tilde{X}^{ij}] = \sum_{i>j} \left(\frac{X^{(i,j)}}{p_{ij}} p_{ij} + 0(1 - p_{ij}) \right) = \sum_{i>j} X^{(i,j)} = B.$$

Furthermore, note that the average number of edges (not including diagonal elements) in B^{sp} is $\mathbb{E}[N_{edges}] = \sum_{i>j} p_{ij}$.

If the p_{ij} are close to 1, then most edges will be included in any realization of B^{sp} and it will inevitably be quite close to B, but not sparse. If the p_{ij} are small, then B^{sp} will be sparse but might be a poor approximation to B in the spectral sense. A good algorithm will choose the p_{ij}'s to ensure both that B^{sp} is close to B and that the number of non-zero edges is small relative to the total unpruned number of edges.

To determine strong sampling probabilities that maintain spectral properties while deleting most edges, we follow Spielman & Srivastava (2011) and first transform B to the identity matrix. Note that $I = B^{-1/2} B B^{-1/2}$, where I is the identity matrix and $B^{-1/2}$ is the matrix that squares to B^{-1} (note that $B^{-1/2}$ is well-defined because B is symmetric and positive definite). Define $\tilde{Y}^{ij} = B^{-1/2} \tilde{X}^{ij} B^{-1/2}$ and $\tilde{I} = \sum_{i>j} \tilde{Y}^{ij} = B^{-1/2} B^{sp} B^{-1/2}$. Note that $\mathbb{E}[\tilde{I}] = I$.

A commonly used formulation of the matrix Chernoff inequality **[1, 73, 74, 90]** bounds the probability that $\tilde{I}$ is far from I. Let M be an upper bound on the $\tilde{Y}^{(i,j)}$'s, so that $0 \leq ||\tilde{Y}^{(i,j)}||_2 \leq M$. Let $\tilde{\lambda}_{min}$ and $\tilde{\lambda}_{max}$ be the minimum and maximum eigenvalues of $\tilde{I}$. For any given $0 < \epsilon < 1$, the matrix Chernoff inequality guarantees that

$$P\left[\tilde{\lambda}_{min} \leq (1 - \epsilon)\right] \leq N \left(e^{-\epsilon^2/2}\right)^{1/M} \text{ and } P\left[\tilde{\lambda}_{max} \geq (1 + \epsilon)\right] \leq N \left(e^{-\epsilon^2/3}\right)^{1/M} \tag{3.7}$$

With this in mind, we can see that a good approximation thus requires that M be small, as the bounds on these probabilities get smaller as M decreases (because $e^{-\epsilon^2/2}$ and $e^{-\epsilon^2/3}$ are smaller than 1). On the other hand, since the sampled pieces are rescaled by $1/p_{ij}$, a sparser approximation (smaller p_{ij}) corresponds to larger M, highlighting the difficulty of balancing M.

For each (i,j), the maximum value that $||\tilde{Y}^{(i,j)}||_2$ takes is $\frac{1}{p_{ij}}||B^{-1/2}X^{(i,j)}B^{-1/2}||$, because it is equal to 0 otherwise. Set

$$\frac{p_{ij}}{K_{deg}} = ||B^{-1/2}X^{(i,j)}B^{-1/2}|| = \mathrm{tr}(B^{-1}X^{(i,j)}) = |w_{ij}|(B_{ii}^{-1} + B_{jj}^{-1} - \mathrm{sign}(w_{ij})2B_{ij}^{-1}), \tag{3.8}$$

for some constant K_{deg}, where the second equality holds since the trace is cyclic and equal to the 2-norm of a rank-1 positive semi-definite matrix. This equalizes the maximum value across $\tilde{Y}^{(i,j)}$, yielding $M = 1/K_{deg}$.

For any given ϵ, ensuring that the probabilities in Eq. 3.7 are small requires that $K_{deg} \geq 4\log(N)/\epsilon^2$ (where the constant 4 is chosen somewhat arbitrarily to ensure small probability for reasonable N, but other values larger than 3 can be chosen with a small alteration in expected number of edges). Thus $K_{deg} = 4\log(N)/\epsilon^2$ guarantees that the eigenvalues of $\tilde{I}$ lie within $[1-\epsilon, 1+\epsilon]$ with high probability (w.h.p.). Consequently, w.h.p., we have

$$(1-\epsilon)\boldsymbol{y}^T\boldsymbol{y} \leq \boldsymbol{y}^T\tilde{I}\boldsymbol{y} \leq (1+\epsilon)\boldsymbol{y}^T\boldsymbol{y} \quad \forall \boldsymbol{y} \in \mathbb{R}^N. \tag{3.9}$$

Given any $\boldsymbol{x} \in \mathbb{R}^N$, set $\boldsymbol{y} = B^{1/2}\boldsymbol{x}$ to see that, w.h.p.,

$$(1-\epsilon)\boldsymbol{x}^T B\boldsymbol{x} \leq \boldsymbol{x}^T B^{sp}\boldsymbol{x} \leq (1+\epsilon)\boldsymbol{x}^T B\boldsymbol{x} \quad \forall \boldsymbol{x} \in \mathbb{R}^N. \tag{3.10}$$

Finally, recalling that $B = -A$ yields the desired approximation.

The average number of edges in the pruned network $\langle N_{edges}\rangle = \sum_{i>j} p_{ij}$. Note that

$$\sum_{i>j} ||B^{-1/2}X^{(i,j)}B^{-1/2}|| = N \qquad \text{(proof in Appendix Section 3.9.1.1).}$$

Hence $\langle N_{edges}\rangle = \sum_{i>j} p_{ij} = NK_{deg}$. Consequently, if $K_{deg} = 4\log(N)/\epsilon^2$ then $\langle N_{edges}\rangle = 4N\log(N)/\epsilon^2$.

As with the sparsification of graph Laplacians **[83]**, for a fixed degree of approximation ϵ, the expected number of edges in A^{sp} need only be $O(N\log(N))$. We now make the strength of this level of sparsity apparent. If the original network is dense, then it has $N(N-1)/2 = O(N^2)$ edges in the symmetric/undirected case (and $N(N-1)$ edges in the directed case). Thus, the expected fraction of edges needed (for a fixed ϵ) goes to 0 as $N \to \infty$. Similarly, if we fix the ratio of edges of A and expected edges of A^{sp}, then the approximation becomes arbitrarily strong (i.e., $\epsilon \to 0$) as $N \to \infty$.

3.5.2. Probabilities from noise-driven covariance. Consider the network of Eq. 3.1 when driven by uncorrelated white noise of variance σ^2 at each node. Set the constant background input $\boldsymbol{b} = 0$ for simplicity (this just shifts the steady-state mean to 0). The covariance matrix C of the resulting dynamics is $C = \mathbb{E}_t[\boldsymbol{x}\boldsymbol{x}^T] - \mathbb{E}_t[\boldsymbol{x}]\mathbb{E}_t[\boldsymbol{x}^T] = \mathbb{E}_t[\boldsymbol{x}\boldsymbol{x}^T]$ and satisfies the Lyapunov equation **[23, 89]**:

$$AC + CA^* = -\sigma^2 I. \tag{3.11}$$

Let A be a normal matrix, meaning $AA^* = A^*A$, where A^* is the conjugate transpose of A (note that every symmetric matrix is indeed normal). Define the symmetrization of A by $A_{symm} = (A + A^*)/2$. It is straightforward to show that $C \propto A_{symm}^{-1}$ (see Appendix Section 3.9.1.2 for details). In particular, if A is symmetric then $C = -\sigma^2 A^{-1}/2$. Substituting $2C/\sigma^2$ for $B^{-1} = -A^{-1}$ in Eq. 3.22 yields

$$p_{ij} = K|w_{ij}|(C_{ii} + C_{jj} - \text{sign}(w_{ij})2C_{ij}), \tag{3.12}$$

with $K = 2K_{deg}/\sigma^2$. In other words, perhaps surprisingly, the pattern of noise-driven correlations exactly encodes the optimal sampling probabilities needed for our application of the matrix Chernoff inequality.

3.6. Numerical Results

In Fig. 2 we show the performance of noise-prune (in the matched diagonal regime) on diagonally-dominant networks with clustered structure (parameters in figure caption). We compare it to a control case in which edges are sampled and either strengthened or pruned (as in Eq. 3.5) but with probabilities just proportional to weight (i.e., without a covariance term and thus without accounting

for higher-order network structure). The proportionality constant for the control is chosen to match the expected number of edges preserved by noise-prune.

The box plots in the first columns of Fig. 2a,b show the distribution of relative change in eigenvalues of the pruned network when compared to the original network, given by $\epsilon_{\lambda_i} = \left|\frac{\tilde{\lambda}_i}{\lambda_i} - 1\right|$, where $\tilde{\lambda}_i$ is the ith eigenvalue of A^{sp}, and λ_i is the ith eigenvalue of A. The box plots in the second column compare the relative change in quadratic forms $\epsilon_{v_i} = \left|\frac{v_i^T A^{sp} v_i}{v_i^T A v_i} - 1\right| = \left|\frac{v_i^T A^{sp} v_i}{\lambda_i} - 1\right|$ for the two approximations, where (v_i, λ_i) is the ith eigenvector-eigenvalue pair of A. Lastly, the box plots in the third column measure how close the eigenvectors of the original network are to being eigenvectors of the pruned network using the normalized dot products of the eigenvectors before and after applying A^{sp}: $\cos(\theta_i) = \frac{\left|v_i^T A^{sp} v_i\right|}{\|A^{sp} v_i\|}$. In all cases, noise-prune performs better than the control, with the performance improving as the networks get larger (panel a vs. b).

We also compare the dynamical response of networks to various inputs before and after pruning. In Fig. 2c we show the response of symmetric clustered networks to random inputs before and after pruning, and find that noise-prune preserves both the responses of individual nodes (left panel) and the network response trajectory as a whole (right panel). We also find similar preservation for structured inputs directed along the slow eigenvectors of the network coupling matrix, which reflect integrative shared dynamical modes that may be used for computation, Fig. 2d. Moreover, noise-prune significantly outperforms the purely weight-based strategy (red vs. blue) and thus using the higher-order structure reflected in the noise covariances dramatically improves the preservation of dynamics in the pruned network.

The theoretical results apply to the case of symmetric matrices but the pruning rule itself is quite general. We thus empirically characterize noise-prune on non-symmetric clustered networks for both random and eigenvector inputs, Fig. 2e and f. Again, noise-prune preserves network dynamics and does much better than a control strategy that relies only on weight, suggesting that good performance extends beyond the theoretical guarantees.

3.7. Extensions

We next briefly describe some extensions of the framework described above (and provide further details in SI).

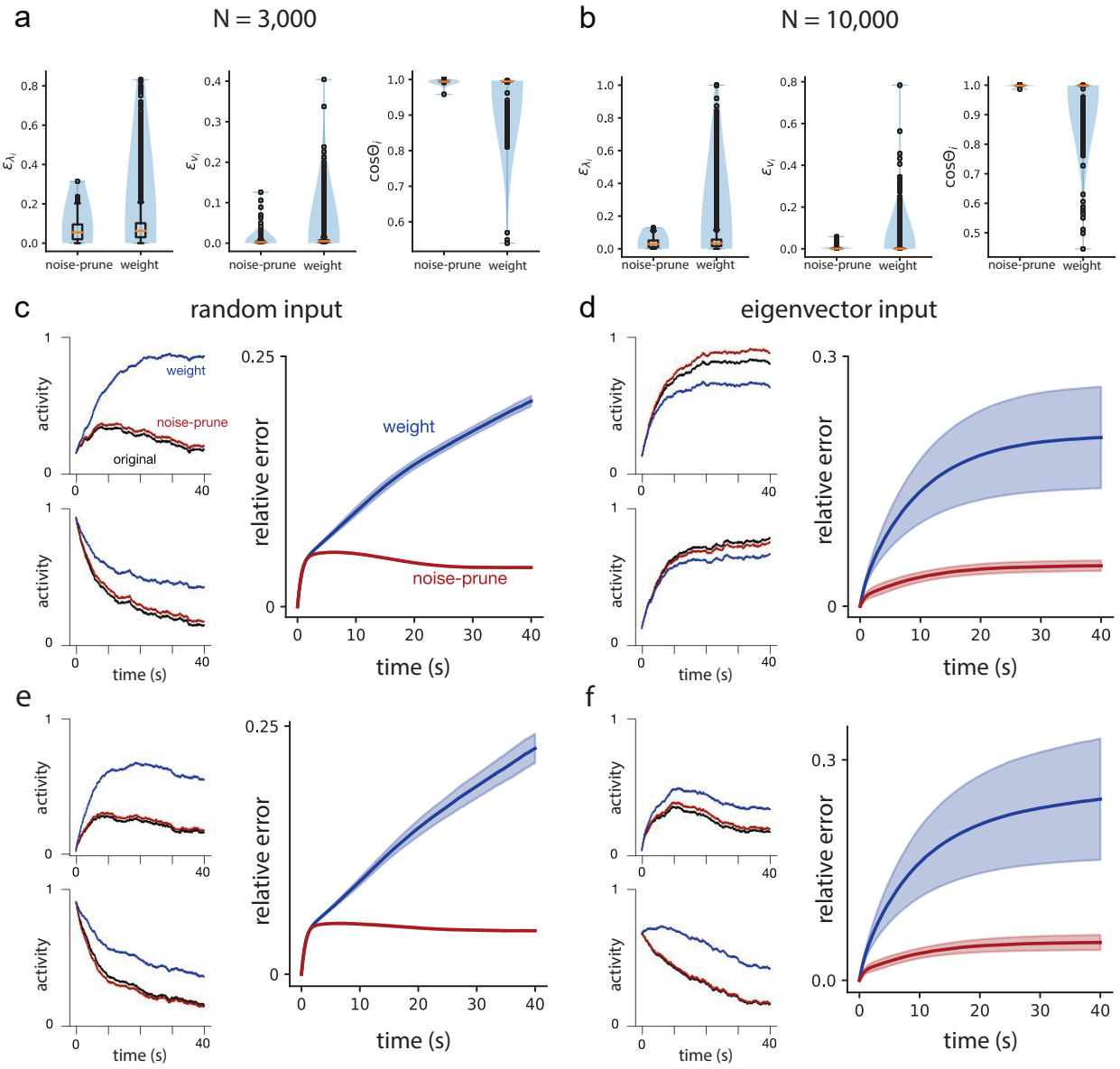

Figure 2: **Noise-prune performance on clustered symmetric and non-symmetric networks.** (a) Performance of noise-prune (left box in each panel) and weight-based pruning (right box in each panel) on networks pruned to 10% density. The left network of size $N = 3,000$ contains 3 clusters of size 100 and 1 cluster of size 2700, with dense within-cluster connections ($60\%, \sim N(1,1)$) and sparse long-range connections (5000 total, $\sim U(0,1)$). From left to right, panels show distribution of ϵ_{λ_i}, ϵ_{v_i} and $\cos(\theta_i)$ (defined in text). Note that good performance corresponds to ϵ_{λ_i} and ϵ_{v_i} near 0 and $\cos(\theta_i)$ near 1. Boxes show upper and lower quartiles, filled circles show outliers, violin plots show density estimate. (b) As in (a) but for larger clustered network ($N = 10,000$, contains 10 clusters of size 100 and 1 cluster of size 9000). (c-f) Dynamical response for networks with three clusters ($1000, 200$, and 800 nodes; connections distributed as in (a)). (caption continued on next page)

Figure 2: Black traces are the original unpruned network; red traces are networks pruned to 20% sparsity with noise-prune in the matched diagonal setting; blue traces are networks pruned to 20% sparsity using probabilities depending solely on weights. (c) Response of symmetric clustered network to random inputs. Panel shows trajectories from dynamical system $\frac{d\boldsymbol{x}}{dt} = A\boldsymbol{x} + \boldsymbol{b} + \boldsymbol{\xi}(t)$, where $\boldsymbol{b}$ is a small constant background input (0.0002), $\boldsymbol{\xi}(t)$ is gaussian white noise, and the initial condition $\boldsymbol{x}(0)$ is chosen with uniformly random entries $U(0,1)$. Left: response of two sample neurons for the three conditions. Right: mean (lines) and standard deviation (shaded area) of relative errors $||\boldsymbol{x}_{orig}(t) - \boldsymbol{x}_{np}(t)||_2/||\boldsymbol{x}_{orig}(t)||_2$ (red) and $||\boldsymbol{x}_{orig}(t) - \boldsymbol{x}_{w}(t)||_2/||\boldsymbol{x}_{orig}(t)||_2$ (blue) over 20 different initial conditions and pruning runs. Here $\boldsymbol{x}_{orig}, \boldsymbol{x}_{np}, \boldsymbol{x}_{w}$ are dynamical responses of the original, noise-pruned, and weight-pruned network respectively. (d) As in (c), but with $\boldsymbol{b} = \boldsymbol{x}(0) = \boldsymbol{v}_i$ where $\boldsymbol{v}_i$ is the eigenvector corresponding to the ith largest eigenvalue of A (or, equivalently, the ith smallest eigenvalue of B). Results averaged over the 20 eigenvectors corresponding to the slowest timescales ($i = 1, \ldots, 20$). (e), (f) are analogous to (c), (d) respectively but for networks with non-symmetric connections.

3.7.1. Approximate probabilities. Our pruning rule is robust to approximate probabilities (as with graph Laplacian sparsification [**83**]). We first discuss underapproximated sampling probabilities first. Recall that the probabilities (a) determine the upper bound M used in Eq. 3.7 and (b) determine $\langle N_{edges} \rangle$ through their sum. Consequently, if some (or all) of the edges are undersampled by a multiplicative factor $\alpha < 1$ (i.e., probabilities $\hat{p}_{ij} = \alpha p_{ij}$ where the p_{ij}'s are the probabilities in Eq. 3.33) then the bound M will be scaled by a factor of $1/\alpha$ and Eq. 3.10 will still hold albeit with a slightly larger $\hat{\epsilon} = \epsilon/\sqrt{\alpha}$, while the pruned network will have fewer edges. In general, if we sample some (or all) edges with a probability higher than the p_{ij}'s, the bound in Eq. 3.10 will remain unharmed; all that will change is simply an increase in the expected number of preserved edges. In particular, any subset of the probabilities can be set to 1 without harming our theoretical guarantees; thus the pruning rule can be naturally applied only to a subset of connections. For more details on these arguments see Appendix Section 3.9.1.4.

3.7.2. Near-diagonally dominant networks. Given a matrix A with eigenvalues λ_i and some constant γ, note that the matrix $A_\gamma = A + \gamma I$ has eigenvalues $\lambda_i + \gamma$ and the same eigenvectors as A. If A is not diagonally-dominant, the application of noise-prune to A can be analyzed by considering its effect on A_γ, with γ chosen large enough that A_γ is diagonally-dominant. There are two additional sources of error in the analysis: first, the probabilities are derived from the covariance matrix of A and are thus sub-optimal for A_γ (but if biological plausability is of no concern, one can simply use the inverse of A_γ to compute p_{ij}'s directly); second, the approximation of Eq. 3.10 holds

for A_γ with some ϵ and the corresponding equation for A includes an additive term of magnitude $\epsilon\gamma$ (see Appendix Section 3.9.2.1 for details).

3.7.3. Rectified linear units. Let $[\cdot]_+ = \max[0, \cdot]$ be a rectified linear activation function and consider the recurrent neural network

$$\frac{d\boldsymbol{x}}{dt} = -D\boldsymbol{x} + [W\boldsymbol{x} + \boldsymbol{b}(t)]_+ . \tag{3.13}$$

As before, define $A = -D + W$. Let A^{sp} be the result of applying noise-prune to A using the probabilities from the linear network defined by A (consequently Eq. 3.15 holds for A, A^{sp}).

Let $\Gamma(t) = \{i : \sum_j W_{ij}x_j + b_j(t) > 0\}$ be the indices of neurons that receive suprathreshold input at time t. Define $A_{\Gamma(t)}$ and $A^{sp}_{\Gamma(t)}$ to be the submatrices produced by removing the rows and columns of A and A^{sp} corresponding to indices not in $\Gamma(t)$ (i.e., those indices corresponding to neurons receiving subthreshold input at time t). The dynamics of the network in Eq. 3.13 are approximately determined by the set of linear systems with connectivity matrices $A_{\Gamma(t)}$, $A^{sp}_{\Gamma(t)}$ (proved in Appendix Section 3.9.2.2). Here, we note that the approximation Eq. 3.15 for A, A^{sp} implies the same approximation for $A_{\Gamma(t)}$ and $A^{sp}_{\Gamma(t)}$ (and show this carefully in SI). Specifically, given some $\Gamma(t)$ with cardinality $|\Gamma(t)|$, let $\Gamma(t, j)$ be the index of the j-th active neuron. Now given $\boldsymbol{y} \in \mathbb{R}^{|\Gamma(t)|}$, define a corresponding $\boldsymbol{x} \in \mathbb{R}^N$ as $\boldsymbol{x}(\Gamma(t, j)) = y(j)$ and remaining entries 0. Then $\boldsymbol{y}^T A^{sp}_{\Gamma(t)}\boldsymbol{y} = \boldsymbol{x}^T A^{sp}\boldsymbol{x}$, and similarly for $A_{\Gamma(t)}$ and A. Substituting into Eq. 3.15 shows that the approximation holds for $A_{\Gamma(t)}$, $A^{sp}_{\Gamma(t)}$.

3.8. Discussion

The structure of the sampling argument, the notion of spectral approximation, and the use of matrix concentration of measure tools was inspired by a rich collection of studies in the realm of graph sparsification [**6**, **83**, **84**], particularly that of the highly influential paper written by Spielman & Srivastava (2011). Our study expands on these results and connects them to neural networks and noisy dynamical systems. In the graph Laplacian context, the counterpart of the comb-cov matrix (see Eq. 3.4) is "effective resistance", which measures the electrical resistance between nodes if the graph is used to model a weighted resistor network. Effective resistance has multiple nice properties [**25**, **83**], such as forming a natural metric between nodes [**47**], and the comb-cov matrix

may be similarly useful for neural networks. There may be further useful connections to be drawn between this set of ideas and noise-driven dynamics in neural networks.

Task-dependent pruning of connections in artificial neural networks has been widely studied in recent years, often with very compelling task-performance results [**7**,**9**,**10**,**17**,**22**,**34**,**36**,**49**,**50**,**61**,**71**]. Current state-of-the-art approaches in machine learning typically train a network to good performance on a task, assign a measure of importance to each connection in the network (typically involving a function of weight and sometimes a measure of impact on the task cost function e.g. elements of the Hessian), remove connections from the network according to this importance measure, and then repeat the cycle of training and pruning until task performance stabilizes with minimal non-zero weights (see [**10**] for a recent review). These iterative train-and-prune algorithms have been incredibly successful in constructing highly sparse networks while approximately preserving task performance. Our work is complementary to these artificial network pruning studies in three ways. First, these studies focus on the supervised, typically feedforward setting, and algorithms are not usually biologically plausible. On the other hand, this study develops an unsupervised, biologically-plausible algorithm for recurrent networks. Second, most existing studies tend to seek good empirical performance in quite challenging real-world applications rather than robust theoretical results, while we focus on developing strong theoretical results in an idealized setting. Ideally, our theoretical foundation will help provide the backbone for a new collection of pruning algorithms. Finally, existing weight pruning algorithms typically do so either based on connection weight or a nonlocal measure of cost function sensitivity. Contrasting this, we combine weight with a local expression that extracts a connection's importance to the network from noisy activity fluctuations. There have been other unsupervised approaches in the literature that are reminiscent of our study, but they merge or remove highly correlated neurons [**3**, **57**, **85**]. The setting, algorithms and theoretical guarantees of these works are noticeably different from ours, and we consider weight pruning rather than removing entire neurons. Note that we do not expect noise-prune to be competitive with state-of-the-art supervised approaches in machine learning when measured by preserving performance on a given task (rather than preserving dynamics). This is natural, as the supervised approaches have the luxury of optimizing their network to perform well on their desired tasks. However, the novel perspective

provided by noise-prune and the theoretical results may be useful in developing more powerful algorithms for task-driven pruning in the future.

The proofs apply to the limited case of symmetric diagonally-dominant linear and rectified linear recurrent networks. While certain networks in the brain may potentially be modeled as diagonally-dominant (e.g., in the high-conductance regime when membrane time constants are very small, indicating high neuronal leak [**15**]), it is unclear how appropriate this approximation will be. More importantly, connections in biological neural networks are not symmetric, as synaptic connections are rarely reciprocal. With that in mind, our undirected framework may apply more naturally to excitatory (or inhibitory) sub-networks with a higher probability of reciprocal connectivity [**82**]. Finally and perhaps most notably, biological networks are highly nonlinear. Thus, while we present some preliminary results for ReLu activiation functions, the theoretical framework presented in this Chapter is far from general. We do provide a further analysis of appropriate nonlinearities in Chapter 4.

Despite these limitations, we highlight two reasons for optimism toward future work. First, in the limited regime where the theory applies, results are very strong and robust (as in graph Laplacian sparsification [**83**, **84**]), asymptotically preserving the entire spectrum of the network, even as the fraction of retained edges goes to 0. Preservation of the entire spectrum is likely stronger than is actually needed for neural networks, which often show redundant coding and low-dimensional dynamics. It may be possible to more weakly approximate a broader family of networks, and this is explored in Chapter 4 as well. Second, the noise-prune rule itself (Eq. 3.4) does not require particular network structure and can in principle be applied to any recurrent network (note that covariance for a general normal matrix is determined by the symmetric part of the matrix). Indeed, we empirically find that noise-prune preserves dynamics in non-symmetric clustered networks, Fig. 2e, f, and thus shows powerful performance beyond the limited regime where theoretical guarantees hold. A more exhaustive empirical characterization of noise-prune is beyond the scope of the present study, but this is a natural direction for future work.

The pruning rule uses randomness in two distinct places. First, it uses noisy fluctuations in neural activity to probe network structure and make global information locally available in the form of activity correlations between pairs of neurons. Second, it randomly decides whether to preserve

(and strengthen) or prune a connection. Randomized algorithms such as noise-prune have found application in a wide domain during recent years, often outperforming deterministic algorithms [**4**,**79**]. This use of randomness in our algorithm is inspired by seemingly ubiquitous noise at multiple levels in neural systems, both internally and externally [**16**, **21**, **48**, **81**]. It remains unclear how much of this "noise" reflects the encoding of unknown variables (i.e., is actually "signal") as opposed to genuine randomness, and to what degree noise is averaged away as opposed to being used as a computational resource. However, randomized algorithms are often appealingly simple, powerful and easy to parallelize, and it is both plausible and widely speculated that brains have evolved to take computational advantage of biological noise [**21**, **59**].

Unlike pruning rules that deterministically remove (typically weak) synapses and simply preserve the others, the synapses targeted by noise-prune are either removed or strengthened, reminiscent of observations that small dendritic spines on neurons (often proxied by synaptic weights in neural models) are highly variable and liable to either vanish or grow and stabilize [**37**]. More generally, a strengthen-or-prune rule like that of our Eq. 3.5 can be applied with different sampling probabilities, which may be appropriate for different settings, and synapses can be strengthened or weakened rather than pruned away entirely. If weights and probabilities are chosen to preserve synaptic weights on average, then the approach approximately preserves total synaptic input to and output from a neuron as well as the dynamics resulting from a given input or network activity state. The theoretical approach may thus be more generally useful in settings where synaptic weight is redistributed across synapses (such as in some homeostatic mechanisms [**91**]). Because total weight mass is preserved by our homeostatic pruning mechanism, there may be interesting work to do in an optimal transport setting as well.

In this study we have focused on pruning synapses while preserving existing network dynamics, thus approaching pruning primarily as homeostatic resource conservation. Pruning in the brain may serve other functions as well, such as making networks faster, better at generalization, or more robust to noise. Given that the pruned network needs to carry out a similar set of input-output transformations to the original network, dynamical patterns being similar between unpruned and pruned networks as proposed in this study could be used as a building block to investigate more complex pruning algorithms that optimize other features of network trajectory. In fact, we continue

to study pruning rules that guarantee similar trajectories of pruned and unpruned networks in Chapter 4.

The approach presented here suggests decomposing into two pieces the difficult problem of learning a sparse network solution to a task. First, a greedy task-driven learning epoch that adds synapses where they might be needed, regardless of efficiency (such as would be expected from correlational [Hebbian] learning processes). Following this, a noisy, task-agnostic, anti-Hebbian epoch during which a subset of synapses enter a labile state and are either consolidated or pruned. The second regime is reminiscent of recent theories of sleep [**51**, **88**]. It would be interesting to attempt to connect sleep phenomenology with unsupervised algorithms such as the one presented in this study.

3.9. Appendix

In this section we expand on the arguments that would have detracted from the flow of the main text. Note that, for completeness, some portions of the main text are repeated here.

3.9.1. Theoretical framework underlying noise-prune. We consider the $N \times N$ coupling matrix of the linear system

$$\frac{dx}{dt} = Ax + \boldsymbol{b}(t), \tag{3.14}$$

and describe how to construct a sparse matrix A^{sp} whose spectrum (and hence dynamics) are similar to A.

To measure the similarity of A and A^{sp}, we adopt the notion of spectral similarity [**83**, **84**] from the field of graph sparsification and require that for some small $\epsilon > 0$,

$$|\boldsymbol{x}^T(A^{sp} - A)\boldsymbol{x}| \leq \epsilon|\boldsymbol{x}^T A\boldsymbol{x}| \quad \forall \boldsymbol{x} \in \mathbb{R}^N. \tag{3.15}$$

The primary theoretical insights of this section are that (a) results on the sparsification of graph Laplacians [**83**, **84**] can be applied, with slight generalization, to pruning signed symmetric diagonally-dominant linear neural networks and (b) that the covariance matrix of the network when driven by noise provides appropriate pruning probabilities. We also discuss what properties of the original network are preserved after sparsifying the matrix A, as well as how these maintained properties are affected when the sampling probabilities are changed.

3.9.1.1. *Sparsification of symmetric, diagonally-dominant networks.* In this section we show how to construct spectral sparsifiers of A. We follow the proof of **[83]**, with some adaptation.

Let A be the coupling matrix of a linear system, as in Eq. 3.1. Note that in order for the linear system to be stable, all the eigenvalues of A must have negative real part (and hence the matrix must be invertible). A non-invertible coupling matrix would correspond to a network with an unrealistically long (i.e., infinite) time-constant.

We impose the further restrictions that A be a symmetric, *diagonally-dominant* matrix; that is, $A_{ij} = A_{ji}$ and $|A_{ii}| \geq \sum_{j \neq i} |A_{ij}|$. In the main text, we focused on the case where this inequality was saturated (i.e., $|A_{ii}| = \sum_{j \neq i} |A_{ij}|$). Here, we expand the proof to include a *strictly diagonally-dominant* A, thus satisfying $|A_{ij}| > \sum_{j \neq i} |A_{ij}|$ (note that the argument is essentially the same and also that both the original and matched diagonal cases of noise-prune simply preserve any excess weight along the diagonal). The diagonal entries of A reflect the intrinsic leak of activity and are negative. Combined with the strict diagonal-dominance requirement, the negative diagonal also implies that the eigenvalues of A are negative, as can be seen from, e.g., a Gershgorin disk argument. Note that the diagonal dominance condition is stronger than the requirement of negative eigenvalues. In the event that A satisfies $|A_{ii}| = \sum_{j \neq i} |A_{ij}|$ as in the main text, invertibility is no longer guaranteed by the Gershgorin disk argument but we assume invertibility based on the stability of the equivalent linear system. We can also relax the invertibility condition by considering the pseudoinverse of A and working in the subspace orthogonal to the nullspace of A (as done for graph Laplacians **[83]**). To sum up, A is negative definite since it is symmetric with negative eigenvalues.

For notational convenience, set $B = -A$ (and note that B is positive definite). The non-zero off-diagonal entries $b_{ij} = -a_{ij} = -w_{ij}$ correspond to the connections in the network (note that throughout w_{ij} refers to the weight in A, i.e., $-b_{ij}$; the argument can be rewritten without introducing B at the cost of extra minus signs).

Edge decomposition For each of these undirected connections (i, j) with $i > j$, we define the edge matrix $X^{(i,j)}$ by

$$X^{(i,j)}_{k\ell} = \begin{cases} |w_{ij}| & \text{if } (k,l) = (i,i) \text{ or } (j,j) \\ -w_{ij} & \text{if } (k,l) = (i,j) \text{ or } (j,i) \\ 0 & \text{otherwise.} \end{cases} \tag{3.16}$$

Note that there is no restriction on the sign of w_{ij}. Also notice that $X^{(i,j)}$ can be written as $\boldsymbol{v}_{ij}\boldsymbol{v}_{ij}^T$ where $\boldsymbol{v}_{ij} \in \mathbb{R}^N$ has ith entry $\sqrt{|w_{ij}|}$ and jth entry $-\operatorname{sgn}(w_{ij})\sqrt{|w_{ij}|}$. Thus $X^{(i,j)}$ a rank-1 matrix. Moreover, since the non-zero eigenvalue is positive, the matrix is positive semidefinite. Also note that specifying $i > j$ above is simply a manner of convention to not double-count connections in the symmetric matrix.

We also define the matrix $X^{(i,i)}$ for all i to have only a single non-zero entry $X^{(i,i)}_{ii} = B_{ii} - \sum_{j\neq i} |w_{ij}|$. Because B is diagonally-dominant with positive diagonal, the single non-zero entry of $X^{(i,i)}$ is positive, again implying that $X^{(i,i)}$ is rank-1 positive semidefinite. We include these diagonal pieces in the sampling argument for completeness but will usually simply treat them as fixed.

The original matrix B is the sum of these edge matrices, $B = \sum_{i\geq j} X^{(i,j)}$ (where the notation $\sum_{i\geq j}$ sums over all existing edge pairs where $i \geq j$).

Sampling edges Now, for $i \geq j$, define the random matrix $\tilde{X}^{ij}$ as

$$\tilde{X}^{ij} = \begin{cases} X^{(i,j)}/p_{ij} & \text{with probability } p_{ij} \\ 0 & \text{otherwise,} \end{cases} \tag{3.17}$$

where $0 < p_{ij} \leq 1$ is some probability we will determine below. Observe that, regardless of the choice of the p_{ij}'s, $\mathbb{E}[\tilde{X}^{ij}] = p_{ij}X^{(i,j)}/p_{ij} = X^{(i,j)}$. Correspondingly, for any set of probabilities, $\boldsymbol{p}$, we can define the matrix $B^{sp,\boldsymbol{p}} = \sum_{i\geq j} \tilde{X}^{ij}$ and we have $\mathbb{E}[B^{sp,\boldsymbol{p}}] = B$ (note that $B^{sp,\boldsymbol{p}}$ will only be sparse if the p_{ij} are small).

Transformation to identity Analogous to Spielman & Srivastava (2011), we implement their argument in our framework by first transforming B into the identity matrix I and finding an appropriate approximation $\tilde{I}$, with the goal of transforming back and arriving at our desired sparsifier

B^{sp}. This step is crucial for preserving the entire spectrum (as required by Eq. 3.15), rather than only the largest eigenvalue (and leads to the comb-cov term in the probabilities).

First observe that $I = B^{-1/2}BB^{-1/2}$, where $B^{-1/2}$ is the matrix whose square is B^{-1} ($B^{-1/2}$ exists since B is invertible and diagonalizable and moreover is real-valued since B is positive definite).[a] Then, defining $Y^{(i,j)} = B^{-1/2}X^{(i,j)}B^{-1/2}$, we have

$$I = B^{-1/2}BB^{-1/2} = \sum_{i \geq j} B^{-1/2}X^{(i,j)}B^{-1/2} = \sum_{i \geq j} Y^{(i,j)}. \tag{3.18}$$

This gives motivation to define the random matrices $\tilde{Y}^{ij} = B^{-1/2}\tilde{X}^{ij}B^{-1/2}$ and $\tilde{I} = \sum_{i \geq j} \tilde{Y}^{ij}$. Note that $E(\tilde{I}) = I$.

Now, for given $0 < \epsilon < 1$, our goal will be to choose p_{ij} in order to guarantee that

$$\boldsymbol{y}^T(1-\epsilon)I\boldsymbol{y} \leq \boldsymbol{y}^T\tilde{I}\boldsymbol{y} \leq \boldsymbol{y}^T(1+\epsilon)I\boldsymbol{y} \quad \forall \boldsymbol{y} \in \mathbb{R}^N, \tag{3.19}$$

with high probability (w.h.p.). If we can do so, then for a given $\boldsymbol{x} \in \mathbb{R}^N$, we can set $\boldsymbol{y} = B^{1/2}\boldsymbol{x}$ in order to arrive at, w.h.p.,

$$\boldsymbol{x}^T(1-\epsilon)B\boldsymbol{x} \leq \boldsymbol{x}^TB^{sp}\boldsymbol{x} \leq \boldsymbol{x}^T(1+\epsilon)B\boldsymbol{x} \quad \forall \boldsymbol{x} \in \mathbb{R}^N, \tag{3.20}$$

where $B^{sp} = B^{-1/2}\tilde{I}B^{-1/2} = \sum_{i \geq j} \tilde{X}^{ij}$, which provides the desired approximation.

Probabilities from matrix Chernoff bound We want our p_{ij} to be as small as possible while still maintaining the inequalities Eq. 3.19, 3.20. To derive good choices for the p_{ij}'s, we apply the matrix Chernoff bound **[1, 73, 74, 90]** to bound the fluctuations of $\tilde{I} = \sum_{i \geq j} \tilde{Y}^{ij}$ around its expectation value, I. Let M be an upper bound on the $\tilde{Y}^{ij}$'s, so that $0 \leq ||\tilde{Y}^{ij}||_2 \leq M$. Let λ_{min} and λ_{max} indicate minimum and maximum eigenvalues. The bound then guarantees that

[a] If the eigenvector decomposition of B is UDU^{-1} then $B^{-1/2}$ can be constructed as $UD^{-1/2}U^{-1}$, where the entries of $D^{-1/2}$ are the inverse square roots of the corresponding entries of D).

$$P\left[\lambda_{min}\left(\sum_{i\geq j}\tilde{Y}^{ij}\right)\leq(1-\epsilon)\right]\leq N\left(\frac{e^{-\epsilon}}{(1-\epsilon)^{(1-\epsilon)}}\right)^{1/M}\leq Ne^{-\epsilon^2/2M}\quad\text{for } 0<\epsilon<1,$$

$$P\left[\lambda_{max}\left(\sum_{i\geq j}\tilde{Y}^{ij}\right)\geq(1+\epsilon)\right]\leq N\left(\frac{e^{\epsilon}}{(1+\epsilon)^{(1+\epsilon)}}\right)^{1/M}\leq Ne^{-\epsilon^2/3M}\quad\text{for } 0<\epsilon \tag{3.21}$$

The hypothesis of the bound requires the spectral norm of the $\tilde{Y}^{ij}$'s to be uniformly bounded across all edges; i.e., $||\tilde{Y}^{ij}||\leq M$. Moreover $||\tilde{Y}^{ij}||$ depends on $1/p_{ij}$, so smaller probabilities lead to a larger bound M. Thus we choose the p_{ij} in order to minimize M.

Since $||\tilde{Y}^{ij}||$ is either $\frac{1}{p_{ij}}\left\|Y^{(i,j)}\right\|$ or 0, choose p_{ij} to equalize the upper bound on $||\tilde{Y}^{ij}||$ across all $i\geq j$:

$$p_{ij}=K_{deg}\left\|Y^{(i,j)}\right\|=K_{deg}||B^{-1/2}X^{(i,j)}B^{-1/2}|| \tag{3.22}$$

where K_{deg} is some constant. This guarantees that $||\tilde{Y}^{ij}||\leq M=1/K_{deg}$. Thus, if we take $K_{deg}\geq 4\log(N)/\epsilon^2$, the probabilities in Eq. 3.21 are guaranteed to be smaller than $1/N$ and $1/N^{1/3}$, respectively. Consequently, this choice of probabilities guarantees that Eqs. 3.19, 3.20 are satisfied w.h.p., as desired.

Note that the constant 4 is chosen somewhat arbitrarily here, with a larger constant corresponding to faster-decaying probabilities in Eq. 3.21 but also a larger number of edges expected to be sampled (since each $\tilde{Y}^{ij}$ is less likely to take on the value of 0).

Bound on number of edges Since edge (i,j) is independently included with probability p_{ij}, the expected number of edges in the network is $\langle N_{edges}\rangle=\sum_{i>j}p_{ij}$ (note the strict inequality here, as $i=j$ does not correspond to edges, but rather the leak in neuronal activity).

We have

$$\sum_{i>j}p_{ij}\leq\sum_{i\geq j}p_{ij}=K_{deg}\sum_{i\geq j}\left\|Y^{(i,j)}\right\|=K_{deg}\sum_{i\geq j}\left\|B^{-1/2}X^{(i,j)}B^{-1/2}\right\| \tag{3.23}$$

Note that $Y^{(i,j)} = \boldsymbol{u}_{ij}\boldsymbol{u}_{ij}^T$, where $\boldsymbol{u}_{ij} = B^{-1/2}\boldsymbol{v}_{ij}$ and $\boldsymbol{v}_{ij}$ is the vector defined after Eq. 3.16. Consequently, $Y^{(i,j)}$ is rank-1 with positive eigenvalue and $\left\|Y^{(i,j)}\right\| = \operatorname{tr} Y^{(i,j)}$. This yields

(3.24)
$$\sum_{i\geq j}\left\|Y^{(i,j)}\right\| = \sum_{i\geq j}\operatorname{tr} Y^{(i,j)} = \operatorname{tr}\left(B^{-1/2}\sum_{i\geq j}X^{(i,j)}B^{-1/2}\right) = \operatorname{tr}(B^{-1/2}BB^{-1/2}) = \operatorname{tr}(I) = N.$$

Thus we have

$$\langle N_{edges}\rangle = \sum_{i>j}p_{ij} = \sum_{i>j}K_{deg}\left\|Y^{(i,j)}\right\| \leq NK_{deg}. \tag{3.25}$$

Simple expression for probabilities Note that $||B^{-1/2}X^{(i,j)}B^{-1/2}|| = \operatorname{tr}\left(B^{-1/2}X^{(i,j)}B^{-1/2}\right) = \operatorname{tr}\left(B^{-1}X^{(i,j)}\right)$, again using the fact that the trace of a positive semi-definite rank-1 matrix is its spectral norm, and that the trace is cyclic (and $B^{-1/2}B^{-1/2} = B^{-1}$ by definition). The product $B^{-1}X^{(i,j)}$ has only two non-zero diagonal terms: its ith diagonal element is given by $|w_{ij}|B_{ii}^{-1} - w_{ij}B_{ij}^{-1}$ and its jth diagonal element is given by $-w_{ij}B_{ji}^{-1} + |w_{ij}|B_{jj}^{-1}$. Using the trivial decomposition $w_{ij} = \operatorname{sgn}(w_{ij})|w_{ij}|$ and adding these two diagonal elements together, we see that

$$p_{ij} = K_{deg}\operatorname{tr}\left(B^{-1}X^{(i,j)}\right) = K_{deg}|w_{ij}|(B_{ii}^{-1} + B_{jj}^{-1} - \operatorname{sgn}(w_{ij})2B_{ij}^{-1}), \tag{3.26}$$

where we note that $B_{ij}^{-1} = B_{ji}^{-1}$, since the inverse of a symmetric matrix is symmetric.

Similarly, the p_{ii} are observed to be

$$p_{ii} = K_{deg}||B^{-1/2}X^{(i,i)}B^{-1/2}|| = K_{deg}\operatorname{tr}\left(B^{-1/2}X^{(i,i)}B^{-1/2}\right) = K_{deg}\operatorname{tr}\left(B^{-1}X^{(i,i)}\right) \tag{3.27}$$

where we again use the cyclic property of the trace. Since the product $B^{-1}X^{(i,i)}$ has only the single non-zero diagonal element $B_{ii}^{-1}(B_{ii} - \sum_{j\neq i}|w_{ij}|)$, we arrive at the simple expression $p_{ii} = K_{deg}B_{ii}^{-1}\left(B_{ii} - \sum_{j\neq i}|w_{ij}|\right)$. Note that in practice we simply set this probability to 1, but include it here for completeness.

Finally, recall that $B = -A$ and note that $A^{sp} = -B^{sp}$ is the outcome of the pruning applied to A. Substituting for B in terms of A, the sampling probabilities are

$$p_{ij} = -K_{deg}|w_{ij}|(A_{ii}^{-1} + A_{jj}^{-1} - \operatorname{sign}(w_{ij})2A_{ij}^{-1}) \tag{3.28}$$

3.9.1.2. *Sampling probabilities from noise-driven covariance.* The matrix inverse term $-A^{-1}$ in Eq. 3.28 has a natural interpretation in terms of the covariance matrix of the corresponding linear dynamical system when driven by white noise. When the network is driven by noise, the dynamics are

$$\frac{dx}{dt} = A\boldsymbol{x} + \sigma\boldsymbol{\xi}(t), \tag{3.29}$$

where $\boldsymbol{\xi}$ is unit variance Gaussian white-noise at each neuron and σ is the standard deviation of the noise (note that this is a stochastic differential equation).

The covariance matrix of the resulting dynamics is given as the solution to the Lyapunov equation [**23**, **89**]:

$$AC + CA^* = -\sigma^2 I. \tag{3.30}$$

Assume that A is normal, meaning that $A^*A = AA^*$, where A^* is the conjugate transpose of A. Note that all symmetric matrices are normal. Since A is normal it can be diagonalized as $A = U\Lambda U^*$, where Λ is a diagonal matrix of eigenvalues and U is unitary.

Substituting the decomposition of A into Eq. 3.30 we have

$$-\sigma^2 I = U\Lambda U^* C + CU\Lambda^* U^* \tag{3.31}$$

so that multiplying this equation through by U^* on the left and U on the right and defining $\tilde{C} = U^*CU$, we arrive at

$$-\sigma^2 I = \Lambda U^* CU + U^* CU\Lambda^* = \Lambda\tilde{C} + \tilde{C}\Lambda^*. \tag{3.32}$$

Since Λ is diagonal, the equation can be solved for the entries of $\tilde{C}$. $\tilde{C}$ is diagonal, with diagonal entries $\tilde{C}_{ii} = -\frac{\sigma^2}{\lambda_i + \lambda_i^*}$, where λ_i and λ_i^* are the i-th diagonal entries of Λ and Λ^* respectively (i.e., the i-th eigenvalue of A). By definition $C = U\tilde{C}U^*$ and thus C has the same eigenvectors as A, with eigenvalues given by the diagonal entries of $\tilde{C}$.

Define the symmetric part of A to be $A_{symm} = \frac{1}{2}(A + A^*)$ and observe that this has eigenvalues $\frac{1}{2}(\lambda_i + \lambda_i^*)$. Thus, $C = -\frac{\sigma^2}{2} A_{symm}^{-1}$. In particular, for the symmetric matrices considered in the

previous section, $C = -\frac{\sigma^2}{2}A^{-1}$. Substituting into the theoretically-derived form for the sampling rule and absorbing $\frac{\sigma^2}{2}$ into the overall constant yields

$$p_{ij} = K|w_{ij}|(C_{ii} + C_{jj} - \text{sign}(w_{ij})2C_{ij}) \tag{3.33}$$

3.9.1.3. *What is preserved.* The notion of spectral sparsification that we adopt from the graph Laplacian literature **[83, 84]** (see Eq. 3.15) is quite strong and here we briefly discuss some of the properties it entails.

Recall that, given $0 < \epsilon < 1$, Eq. 3.20 guarantees that

$$\boldsymbol{x}^T(1-\epsilon)B\boldsymbol{x} \leq \boldsymbol{x}^T B^{sp}\boldsymbol{x} \leq \boldsymbol{x}^T(1+\epsilon)B\boldsymbol{x} \quad \forall \boldsymbol{x} \in \mathbb{R}^N, \tag{3.34}$$

so that substituting $A = -B$ and rearranging yields the approximation from the main text

$$|\boldsymbol{x}^T(A^{sp} - A)\boldsymbol{x}| \leq \epsilon|\boldsymbol{x}^T A\boldsymbol{x}| \quad \forall \boldsymbol{x} \in \mathbb{R}^N, \tag{3.35}$$

where we use the fact that A is negative definite to see that $-\boldsymbol{x}^T A\boldsymbol{x} = |\boldsymbol{x}^T A\boldsymbol{x}|$.

By definition, Eq. 3.35 approximately preserves A as a quadratic form and thus apart from the eigenvalues and products described below, it also preserves properties of the dynamical system that depend on A as a quadratic form, such as the resting state variances, the diagonal elements of A and the combination-of-covariances (comb-covs).

Eigenvalues Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ and $\tilde{\lambda}_1 \leq \tilde{\lambda}_2 \leq \cdots \leq \tilde{\lambda}_N$ be the eigenvalues of B and B^{sp} respectively.

Let S denote the collection of subspaces $U \subset \mathbb{R}^N$ with $\dim U = k$, and consider the functions $f_B, f_{B^{sp}} : S \to \mathbb{R}$ given by

$$f_B(U) = \max_{\substack{\boldsymbol{x}\in U \\ \|\boldsymbol{x}\|=1}} \boldsymbol{x}^T B\boldsymbol{x}, \qquad f_{B^{sp}}(U) = \max_{\substack{\boldsymbol{x}\in U \\ \|\boldsymbol{x}\|=1}} \boldsymbol{x}^T B^{sp}\boldsymbol{x}. \tag{3.36}$$

Let $U \in S$ be a given subspace of $\mathbb{R}^N$ with dimension k. Since $(1-\epsilon)\boldsymbol{x}^T B\boldsymbol{x} \leq \boldsymbol{x}^T B^{sp}\boldsymbol{x} \leq (1+\epsilon)\boldsymbol{x}^T B\boldsymbol{x}$ for all $\boldsymbol{x} \in \mathbb{R}^N$, we can take the maximum over all $\boldsymbol{x} \in U \subset \mathbb{R}^N$ with unit norm to see that

$$(1-\epsilon)f_B(U) \leq f_{B^{sp}}(U) \leq (1+\epsilon)f_B(U). \tag{3.37}$$

Since this inequality holds for any subspace, taking a minimum over all subspaces in S still preserves the inequality:

$$(1-\epsilon)\min_{U\in S} f_B(U) \leq \min_{U\in S} f_{B^{sp}}(U) \leq (1+\epsilon)\min_{U\in S} f_B(U). \tag{3.38}$$

Thus, by the Courant-Fischer Theorem, we arrive at

$$(1-\epsilon)\lambda_k \leq \tilde{\lambda}_k \leq (1+\epsilon)\lambda_k, \quad \forall 1 \leq k \leq N \tag{3.39}$$

Thus all eigenvalues are preserved within a multiplicative factor of ϵ.

Eigenvectors Here we show that the angle between eigenvectors is preserved up to a factor depending on the arbitrarily small degree of spectral approximation ϵ. First, note that a rearrangement of the Davis-Kahan theorem states

$$\sqrt{1-\frac{4||A-A^{sp}||^2}{\delta_i^2}} \leq \cos\angle(v_i,\tilde{v}_i), \qquad \text{for all } i \tag{3.40}$$

where v_i and $\tilde{v}_i$ are the ith eigenvectors of A and A^{sp} respectively, and

$$\delta_i = \min_{j:j\neq i} |\lambda_i(A) - \lambda_j(A)| > 0. \tag{3.41}$$

Fix $\gamma > 0$. Now, setting $\epsilon = \frac{\gamma}{2\lambda_{max}}$ and constructing the corresponding A^{sp}, we know

$$||A-A^{sp}|| = \sup_{||x||=1} |x^T(A-A^{sp})x| \leq \sup_{||x||=1} \epsilon|x^TAx| = \epsilon\lambda_{max} = \frac{\gamma}{2} \tag{3.42}$$

where the first equality holds since $A - A^{sp}$ is Hermitian. Thus, we have

$$\sqrt{1-\frac{\gamma^2}{\delta_i^2}} \leq \sqrt{1-\frac{4||A-A^{sp}||^2}{\delta_i^2}} \leq \cos\angle(v_i,\tilde{v}_i). \tag{3.43}$$

Since $\gamma > 0$ was arbitrary, this quantity can be made arbitrarily close to 1. That is, we can guarantee that corresponding eigenvectors of A and A^{sp} point in nearly the same direction.

Preserved matrix-vector products First, note that there exist N linearly independent vectors $\{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_N\}$ (i.e., a basis for $\mathbb{R}^N$) for which the matrix vector products are preserved between B and B^{sp} (or A and A^{sp}) to within ϵ. Let $\boldsymbol{v}_k$ be an eigenvector of $\tilde{I}$ with eigenvalue $1+\delta_k$ (note that, from Eq.

3.39, $|\delta_k| \leq \epsilon$). Define $\boldsymbol{w}_k = B^{-1/2}\boldsymbol{v}_k$ and note that $B\boldsymbol{w}_k = B^{1/2}\boldsymbol{v}_k$. Now $B^{sp}\boldsymbol{w}_k = B^{1/2}\tilde{I}B^{1/2}\boldsymbol{w}_k = B^{1/2}\tilde{I}\boldsymbol{v}_k = (1+\delta_k)B^{1/2}\boldsymbol{v}_k = (1+\delta_k)B\boldsymbol{w}_k$. Consequently, $||(B - B^{sp})\boldsymbol{w}_k|| \leq \epsilon||B\boldsymbol{w}_k||$.

Second, note that the eigenspaces of B and B^{sp} are close, as we show empirically in Fig. 2 of the main text, though precise bounds will depend on how close the corresponding eigenvalue is to another eigenvalue in the spectrum.

Finally, scalar concentration of measure arguments suggests that a rule of the form in Eq. 3.5 should preserve dense matrix-vector products, provided the entries in the matrix do not grow too large (as for the matrix concentration of measure case). Note, however, that the products of B and B^{sp} with sparse vectors may be quite different (as will be true for any sparse matrix approximation), because these products are determined by the sum of only a few entries in B and B^{sp}.

3.9.1.4. *Partial sampling and robustness to changing probabilities.*

Oversampling. The Chernoff bound in Eq. 3.21 depends on the sampling probabilities only through the upper bound on the norm of the edge matrices, requiring $0 \leq ||Y_{ij}|| \leq 1/K_{deg}$. In particular, if the derived p_{ij} for an edge is < 1 then the same bound holds for any $\tilde{p}_{ij} \geq p_{ij}$ (see below for $p_{ij} > 1$). Consequently, the probabilities described in Eq. 3.22 above are a lower bound, for a given desired degree of approximation (ϵ). If some of the edges are sampled with a greater probability than the theoretical result derived, the approximation equation Eq. 3.22 will still hold (though some of the terms in the sum of Eq. 3.18 will have norm less than $1/K_{deg}$).

The only consequence of over-sampling synapses is that the number of connections in the pruned network will be greater, but the increase is as well behaved as could be desired, corresponding exactly to the degree of over-sampling as $\langle N_{edges} \rangle = \sum_{i>j} p_{ij}$. Furthermore, there is no harm in setting all of the p_{ii} to 1, as these probabilities do not correspond to edges, but rather the diagonal terms, which relate to the intrinsic leak in the activity of neurons in Eq. 3.1.

Moreover, as long as the sampling probabilities are above the theoretically-derived bound, they can be chosen completely independently at each synapse and do not need to compensate for each other in any way.

Misspecified probabilities. Sampling-based sparsification is quite robust to misspecified sampling probabilities [**83**]. Again, this robustness emerges because probabilities only affect the

Chernoff bound through their effect on the norm of the edge matrices. If some synapses are undersampled using probability $\hat{p}_{ij} = \alpha p_{ij}$, with $\alpha < 1$, the bound on the $||\tilde{Y}^{ij}||$'s inflates by a factor of $1/\alpha$ and the degree of approximation becomes $\hat{\epsilon} = \epsilon/\sqrt{\alpha}$ while preserving the same bound on the probabilities in Eq. 3.21 (thus maintaining the likelihood of our approximation occuring w.h.p.). To see this, observe that

$$P\left[\lambda_{min} \leq (1-\hat{\epsilon})\right] \leq N\left(e^{-\hat{\epsilon}^2/2}\right)^{\alpha/M} = N\left(e^{-\epsilon^2/2}\right)^{1/M}, \tag{3.44}$$

and similarly for the other inequality in Eq. 3.21. Note here that we could instead choose to maintain our original degree of approximation ϵ, but this would correspond to the larger upper bound

$$P\left[\lambda_{min} \leq (1-\epsilon)\right] \leq N\left(e^{-\epsilon^2/2}\right)^{\alpha/M} \tag{3.45}$$

which means that Eq. 3.20 would occur with lower probability.

Fixed edges. The sampling argument can be applied to only a subset of edges in several ways. A particularly natural approach is to simply set the sampling probability for a fixed edge $\tilde{p}_{ij} = 1$ and note that, if $p_{ij} < 1$ the bound $||\tilde{Y}^{ij}|| \leq M$ still holds and so do the subsequent theoretical results.

A second way to apply the argument to a subset of edges is to write the matrix A as $A_{fixed} + A_{sample}$, where A_{fixed} is the submatrix of edges that are to be preserved and A_{sample} is the submatrix of edges to be either pruned or strengthened. The argument in Section 3.9.1.1 can then be applied to A_{sample} (note that A_{sample} is diagonally dominant and positive semidefinite). This formulation has the disadvantage that the predicted sampling probabilities depend on the covariance matrix determined by A_{sample} rather than A, but this covariance matrix may be natural in certain contexts.

Furthermore, while the diagonal terms sampled with probability p_{ii} do not correspond to edges, we can still fix them with no harm to our theoretical results (i.e., set $p_{ii} = 1$ as noted earlier in the oversampling paragraph of this section).

Synapses with probabilities greater than 1. A calculated probability term for a synapse p_{ij} that is > 1 can be handled in two ways. One solution is to convert each synaptic weight into pieces with predicted probability < 1 and rewrite the sum in Eq. 3.18 as involving multiple pieces corresponding to the edge (i, j) each sampled with probability 1. Note that this will increase K_{deg} slightly but does not change the actual form of the sampling rule (the edge is just preserved). A

second approach is to split the matrix into a deterministic and a sampled piece, and apply the argument to the sampled piece (as in the argument for fixed edges above). Again this has the drawback that the predicted sampling probabilities would not be given by the covariance matrix of the entire network.

3.9.2. Extensions.

3.9.2.1. *Near-diagonally-dominant networks.* Let the matrix A be a (not necessarily diagonally-dominant) symmetric negative definite matrix corresponding to the coupling matrix of a linear system such as in Eq. 3.1. We analyze the effect of applying noise-prune to A in terms of its distance from a diagonally dominant matrix.

As before, we let $B = -A$ and analyze the effect of the rule on B. Note that the noise-driven matrix of the linear system $C \propto -A^{-1} = B^{-1}$, and that the sampling probabilities yielded by noise-prune are $p_{ij} = K|w_{ij}|(C_{ii} + C_{jj} - 2\,\mathrm{sgn}(w_{ij})C_{ij}) = K_{deg}|w_{ij}|(B^{-1}_{ii} + B^{-1}_{jj} - 2\,\mathrm{sgn}(w_{ij})B^{-1}_{ij})$ for $i > j$. We will also set any excess diagonal probabilities $p_{ii} = 1$ (note that this is implicitly done in both the original and matched diagonal settings of noise-prune in the main paper).

Set $\gamma > 0$ and define the matrix $B_\gamma = B + \gamma I$. Let B have eigenvalues λ_k and eigenvectors $\boldsymbol{v}_k$ and observe that B_γ has eigenvalues $\lambda_i + \gamma$ and the same eigenvectors as B. Moreover, note that applying noise-prune to B with some set of probabilities to yield B^{sp} is equivalent to applying noise-prune to B_γ with the same set of probabilities to yield $B_{sp=B^{sp}+\gamma I}$ (though these are not the optimal probabilities for B_γ).

Now take γ large enough so that B_γ is diagonally dominant (the approximation described below will be good if γ is small). The framework described in Section 3.9.1.1 can then be applied to B_γ and the probabilities saturating the Chernoff bound are $p^{(\gamma)}_{ij} = K_{deg}|w_{ij}|([B_\gamma]^{-1}_{ii} + [B_\gamma]^{-1}_{jj} - 2\,\mathrm{sgn}(w_{ij})[B_\gamma]^{-1}_{ij})$.

In particular, note that

$$[B_\gamma]^{-1}_{ij} = \left(\sum_k \frac{1}{\lambda_k + \gamma}\boldsymbol{v}_k\boldsymbol{v}_k^T\right)_{ij} = \sum_k \frac{1}{\lambda_k + \gamma}\left(\boldsymbol{v}_k\boldsymbol{v}_k^T\right)_{ij} = \sum_k \frac{1}{\lambda_k + \gamma}(\boldsymbol{v}_k)_i(\boldsymbol{v}_k)_j \quad \forall i, j \tag{3.46}$$

and similarly

$$B_{ij}^{-1} = \left(\sum_k \frac{1}{\lambda_k} \boldsymbol{v}_k \boldsymbol{v}_k^T\right)_{ij} = \sum_k \frac{1}{\lambda_k} \left(\boldsymbol{v}_k \boldsymbol{v}_k^T\right)_{ij} = \sum_k \frac{1}{\lambda_k} (\boldsymbol{v}_k)_i (\boldsymbol{v}_k)_j \quad \forall i, j, \tag{3.47}$$

where $(\boldsymbol{v}_k)_\ell$ denotes the ℓth entry of the kth eigenvector $\boldsymbol{v}_k$. Then we can see that, for $i > j$,

$$\begin{aligned} p_{ij}^{(\gamma)} &= K_{deg} |w_{ij}| \sum_k \frac{1}{\lambda_k + \gamma} \left((\boldsymbol{v}_k)_i^2 + (\boldsymbol{v}_k)_j^2 - 2\,\mathrm{sgn}(w_{ij})(\boldsymbol{v}_k)_i (\boldsymbol{v}_k)_j\right) \\ &= K_{deg} |w_{ij}| \sum_k \frac{1}{\lambda_k + \gamma} ((\boldsymbol{v}_k)_i - \mathrm{sgn}(w_{ij})(\boldsymbol{v}_k)_j)^2 \\ &\leq K_{deg} |w_{ij}| \sum_k \frac{1}{\lambda_k} ((\boldsymbol{v}_k)_i - \mathrm{sgn}(w_{ij})(\boldsymbol{v}_k)_j)^2 \\ &= p_{ij}, \end{aligned}$$

where the inequality follows from the fact that $\frac{1}{\lambda_k + \gamma} \leq \frac{1}{\lambda_k}$ and the rest of the terms in the expression are all nonnegative.

Thus $p_{ij}^{(\gamma)} \leq p_{ij}$ for all $i \geq j$ and sparsifying B_γ using the probabilities p_{ij} yields Eq. 3.20 for at most the same degree of error ϵ we would get if we used the $p_{ij}^{(\gamma)}$'s instead (see Section 3.9.1.4 for more details on oversampling). That is,

$$(1 - \epsilon) \boldsymbol{x}^T B_\gamma \boldsymbol{x} \leq \boldsymbol{x}^T (B_\gamma)^{sp} \boldsymbol{x} \leq (1 + \epsilon) \boldsymbol{x}^T B_\gamma \boldsymbol{x} \quad \forall \boldsymbol{x} \in \mathbb{R}^N. \tag{3.48}$$

Now observing that $(B_\gamma)^{sp} = B^{sp} + \gamma I$ allows us to subtract $\boldsymbol{x}^T \gamma I \boldsymbol{x}$ through our inequality to arrive at

$$(1 - \epsilon) x^T B \boldsymbol{x} - \epsilon\gamma \boldsymbol{x}^T \boldsymbol{x} \leq \boldsymbol{x}^T B^{sp} \boldsymbol{x} \leq (1 + \epsilon) \boldsymbol{x}^T B \boldsymbol{x} + \epsilon\gamma \boldsymbol{x}^T \boldsymbol{x} \quad \forall \boldsymbol{x} \in \mathbb{R}^N. \tag{3.49}$$

In other words, sparsifying B using the same probabilities we used for B_γ guarantees a result similar to that of Eq. 3.20, but with an additional additive error of $\epsilon\gamma \boldsymbol{x}^T \boldsymbol{x}$. In particular, if $\boldsymbol{x}$ has unit norm then the additive error is simply $\epsilon\gamma$.

3.9.3. Rectified linear units. Define the rectified linear activation function $[\cdot]_+ = \max[0, \cdot]$ and consider the recurrent neural network

$$\frac{d\boldsymbol{x}}{dt} = -D\boldsymbol{x} + [W\boldsymbol{x} + \boldsymbol{b}(t)]_+. \tag{3.50}$$

As before, define $A = -D + W$, and let A^{sp} be the result of applying noise-prune to A using the probabilities from the linear network defined by A (so that Eq. 3.20 holds for A and A^{sp}).

Let $\Gamma(t) = \{i : \sum_j W_{ij}x_j + b_j(t) > 0\}$ be the indices of neurons that receive suprathreshold input at time t. Define $A_{\Gamma(t)}$ and $A^{sp}_{\Gamma(t)}$ to be the submatrices produced by removing the rows and columns of A and A^{sp} corresponding to indices not in $\Gamma(t)$. We will show that the dynamics of the network in Eq. 3.50 are approximately determined by the set of linear systems (indexed by t) with coupling matrices $A_{\Gamma(t)}, A^{sp}_{\Gamma(t)}$. In other words, the dynamics of a rectified linear network switch among the dynamics of a set of linear networks, with the appropriate linear network at a moment in time determined by the subset of neurons that receive suprathreshold input (see **[31, 32]** for more on this argument).

For convenience, let $\Gamma(t)^c$ be the complement of $\Gamma(t)$; that is, $\Gamma(t)^c$ is the collection of neurons that receive zero input. The neurons in $\Gamma(t)^c$ either have zero activity (and thus can be ignored) or have nonzero activity but receive zero input (and thus contribute feedforward input to the rest of the network that can be absorbed into the input vector). Define $\boldsymbol{x}_{\Gamma(t)}$ and $\boldsymbol{b}_{\Gamma(t)}$ to be the vectors produced by removing the entries of $\boldsymbol{x}$ and $\boldsymbol{b}$ corresponding to the indices in $\Gamma(t)^c$, as well as $\boldsymbol{x}_{\Gamma(t)^c}$ to be the vector produced by removing the entries of $\boldsymbol{x}$ corresponding to the indices in $\Gamma(t)$. Lastly, define $\delta\boldsymbol{b}_{\Gamma(t)}$ to be the feedforward contribution of $\boldsymbol{x}_{\Gamma(t)^c}$ (more precisely, the ith entry of this vector is given by $\sum_{j\in\Gamma(t)^c} W_{ij}x_j$ with $i \in \Gamma(t)$ listed in increasing order) that we will absorb into the new input vector for our smaller system in $\mathbb{R}^{|\Gamma(t)|}$, defined to be $\tilde{\boldsymbol{b}}_{\Gamma(t)} = \boldsymbol{b}_{\Gamma(t)} + \delta\boldsymbol{b}_{\Gamma(t)}$.

Now the dynamics of the network in some small time interval around t are determined by the linear system,

$$\frac{d\boldsymbol{x}_{\Gamma(t)}}{dt} = A_{\Gamma(t)}\boldsymbol{x}_{\Gamma(t)} + \tilde{\boldsymbol{b}}_{\Gamma(t)}. \tag{3.51}$$

And the nodes in $\Gamma(t)^c$ either have 0 activity or are decaying to 0 with the leak time constant.

Note that Eq. 3.20 holds for $A_{\Gamma(t)}, A^{sp}_{\Gamma(t)}$ as well. Let $\Gamma(t,j)$ be the index of the j-th active neuron at time t. Given $\boldsymbol{x}_{\Gamma(t)} \in \mathbb{R}^{|\Gamma(t)|}$, consider the natural extension vector $\boldsymbol{x} \in \mathbb{R}^N$ whose entry in $\Gamma(t,j)$ is the j entry of $\boldsymbol{x}_{\Gamma(t)}$ and whose entries in $\Gamma(t)^c$ are 0. Then $\boldsymbol{x}^T_{\Gamma(t)} A^{sp}_{\Gamma(t)} \boldsymbol{x}_{\Gamma(t)} = \boldsymbol{x}^T A^{sp} \boldsymbol{x}$ (and similarly, $\boldsymbol{x}^T_{\Gamma(t)} A_{\Gamma(t)} \boldsymbol{x}_{\Gamma(t)} = \boldsymbol{x}^T A \boldsymbol{x}$), so the fact that Eq. 3.20 holds for A, A^{sp} implies that it holds for $A_{\Gamma(t)}, A^{sp}_{\Gamma(t)}$ (for all t). Thus, among other quantities, the spectrum of $A_{\Gamma(t)}$ is approximately preserved (to within ϵ) by $A^{sp}_{\Gamma(t)}$. Thus, we see that noise-prune preserves the dynamics of linear systems described the submatrices $A_{\Gamma(t)}$. Finally, $\tilde{\boldsymbol{b}}_{\Gamma(t)}$ depends on the weights through $\delta \boldsymbol{b}_{\Gamma(t)}$, which may be perturbed in the sp system, though it is preserved in expectation. However, perturbations are likely to be small because this additional feedforward input comes from the small subset of low-activity neurons in $\boldsymbol{x}_{\Gamma(t)^c}$ that receive sub-threshold input and are approaching zero activity but have not completely decayed yet (which they do so with time-constant given by the leak). In short, the dynamics of a rectified linear network are approximately preserved when its coupling matrix is sparsified in the same manner as that of a linear network.

CHAPTER 4

Unrestricted Pruning and Error Bounds for Pruned Nonlinear Networks

4.1. Abstract

Network sparsification via weight pruning is a technique used to reduce model size and computation time, ideally while maintaining performance. Matrix concentration of measure inequalities are powerful sparsification tools, but are currently underutilized in neural network literature due to their recent emergence. We present a sparsification algorithm using the matrix Bernstein inequality, vastly reducing the number of non-zero entries of rectangular weight matrices while approximately preserving their spectrum. This algorithm applies to rectangular matrices in general, an improvement on the idealized conditions of the results in Chapter 3. Following this result, we produce an analytic argument that bounds the error between a pruned network's output and that of its original, dense counterpart. Under mild conditions such as sigmoid activation functions, this bound is a linear combination of the layer-wise spectral errors guaranteed by our sparsification algorithm. This error bound also applies to any activation layer, not only the output of a network; thus, the bound is useful in the context of both feed-forward neural networks and discrete-time recurrent neural networks.

4.2. Introduction

Neural network pruning has a rich history dating back to the 1980s, originally motivated by the desire to reduce network complexity while empirically maintaining performance [**36**, **49**]. As networks with heavy storage requirements and computationally intensive structures have become increasingly popular, a recent surge of empirical pruning work has appeared in the literature [**7**,**9**,**10**,**17**,**22**,**34**,**50**,**61**,**71**]. In Frankle and Carbin's highly influential empirical work of 2019, they conjectured the Lottery Ticket Hypothesis (LTH). The LTH claimed the existence of subnetworks that, when trained in isolation of the original randomly initialized network, can match test accuracy

comparable to that of the original network after training for similar number of training epochs [**22**]. Following this, Ramanujan et al.'s empirical pruning study led to their conjecture of the aptly named strong-LTH, which postulated the existence of such subnetworks, but included the additional assumption that they need not be further trained to reach these comparable accuracy levels [**69**].

Several theoretical pruning studies have emanated from these conjectures in the last few years, each providing pruning algorithms with theoretical guarantees on the performance of the pruned network, as well as bounds on the number of connections in the network. Namely, Malach et al. proved (a technical formulation of) the strong-LTH in 2020 for fully-connected, randomly initialized networks with ReLu activation functions [**56**]. They show that a ReLu network with ℓ layers can be well-approximated by a subnetwork of a random network with 2ℓ layers; they first "over-parametrize", doubling the number of layers of their network, then prune these layers deterministically to arrive at a "sparse" network that approximates the original network arbitrarily well. However, these networks are still quite dense, and are twice as deep as the original networks. Lastly, there is another line of theoretical work that involves sampling-based pruning schemes for neural networks, but they each use scalar concentration of measure inequalities to sample their networks' parameters [**7**, **8**, **53**].

Contrasting the work described above, we present a pruning algorithm largely powered by the powerful matrix-valued Bernstein inequality [**70**, **90**]. This result comes from a family of matrix concentration inequalities that have been used in a variety of applications such as matrix completion and sparsification, semi-definite program solvers, and approximate eigenvector computation [**18**]. However, these matrix-valued concentration of measure inequalities have yet to make their way into the neural network pruning literature, despite providing powerful theoretical guarantees about preserved matrix spectral norm. Unlike the results in Chapter 3 [**60**], which were limited to square, symmetric, diagonally-dominant weight matrices, the matrix Bernstein inequality applies to general rectangular matrices. We then show that applying this matrix pruning algorithm to each weight matrix of a feed-forward neural network provides an output error bound between the pruned network and the original that can be made arbitrarily small under mild conditions (such as sigmoid activation functions and unitary weight matrices). Finally, we discuss these error bounds in the context of discrete-time recurrent neural networks.

4.3. Problem Setup

Let $G(\boldsymbol{x}; \boldsymbol{\Theta})$ be a feed-forward neural network with activation function $\boldsymbol{g}(\cdot)$ applied after each layer. The input of this network is $\boldsymbol{x} \in \mathbb{R}^{N_0}$, and it has parameters

$$\boldsymbol{\Theta} = (W^{(1)}, W^{(2)}, \ldots, W^{(n)}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}, \ldots, \boldsymbol{b}^{(n)}),$$

where $W^{(\ell)} \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and $b^{(\ell)} \in \mathbb{R}^{N_\ell}$. That is, the activity at layer ℓ is given by

$$\boldsymbol{x}^{(\ell)} = \boldsymbol{g}(W^{(\ell)} \boldsymbol{x}^{(\ell-1)} + \boldsymbol{b}^{(\ell)}) \quad \text{for } \ell = 1, \ldots, n,$$

so that $\boldsymbol{x}^{(n)} = G(\boldsymbol{x}^{(0)}; \boldsymbol{\Theta}) \in \mathbb{R}^{N_n}$.

We are seeking a pruning algorithm that can sparsify each layer's weight matrix, yielding a network $G(\boldsymbol{x}; \boldsymbol{\Theta}^{\boldsymbol{sp}})$ with parameters

$$\boldsymbol{\Theta}^{sp} = (W^{(1,sp)}, W^{(2,sp)}, \ldots, W^{(n,sp)}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}, \ldots, \boldsymbol{b}^{(n)}).$$

In addition, we want the sparsified network to still transform inputs in a manner similar to its original dense counterpart. In other words, we desire

$$\left\| G(\boldsymbol{x}^{(0)}; \boldsymbol{\Theta}) - G(\boldsymbol{x}^{(0)}; \boldsymbol{\Theta}^{\boldsymbol{sp}}) \right\|_2 \leq \epsilon, \tag{4.1}$$

where ϵ is small. By utilizing sparse weight matrices, both computation time and memory requirements of the network can be vastly reduced with minimal impact on performance.

Our algorithm, largely inspired by that of Drineas and Zouzias (2011), will guarantee that each layer's sparsification $W^{(\ell,sp)}$ has $O(n \log n \left\|W^{(\ell)}\right\|_F^2 \delta_\ell^{-2})$ expected edges, where $n = \max\{N_\ell, N_{\ell-1}\}$ and $\delta_\ell > 0$. At the same time, it guarantees the linear operator bound

$$\left\| W^{(\ell)} - W^{(\ell,sp)} \right\|_2 \leq \delta_\ell$$

for each $\ell = 1, \ldots, n$. Here, each δ_ℓ can be made arbitrarily small, at the cost of an increase in expected remaining edges (which grows inversely proportional to δ_ℓ^2). However, caution should be exercised when decreasing δ_ℓ; if $\left\|W^{(\ell)}\right\|_F$ is large, the sparsification guarantee could hold trivially (see Appendix [Section 4.9.2]).

Our next contribution will show that, once we have our hands on these layer-wise linear bounds, we can convert them into a network-level nonlinear bound of the form Eq. 4.1, using only the additional assumption that the nonlinear activation function is a *contraction* (see Appendix [Section 4.9.1] for our definition of contractions, which includes traditionally popular neural network activation functions such as sigmoids, ReLu, etc.). This argument, while technically cumbersome, is rooted in the simple idea that the error between two objects is made smaller when these objects are passed through a contraction. As such, repeat applications of (sparsified) affine transformations followed by contraction nonlinearities only mildly distort network input. The overall degree of approximation ϵ is simply a linear combination of each δ_ℓ, and thus can also be made as small as desired. Perhaps surprisingly, if we consider a network with sigmoid activation function and unitary weight matrices, ϵ is merely a scaled sum of all δ_ℓ (see Corollary 4.8 for the details of this special case).

4.4. An Unrestricted Pruning Rule for Rectangular Matrices

We will focus on the development of a pruning algorithm used to sparsify a single rectangular matrix (and in Section 4.6, we will apply this pruning algorithm to each of the weight matrices of a network). That is, given a matrix $A \in \mathbb{R}^{M\times N}$ and a desired degree of spectral approximation $\delta > 0$, we seek to construct a sparse matrix $A^{sp} \in \mathbb{R}^{M\times N}$ satisfying $\|A - A^{sp}\|_2 < \delta$. We begin by stating the sparsification algorithm below. Note that we use the notation $[n] = \{1, \dots, n\}$ for $n \in \mathbb{N}$.

Algorithm 3 Rectangular Bernstein Matrix Sparsification Algorithm

Input: $A \in \mathbb{R}^{M\times N}$ and error tolerance $\delta > 0$.

1. **Set** $B = A$ and set $B_{ij} = 0$ if $A_{ij} \leq \delta/2n$.
2. **Set** $s = \frac{128}{3\delta^2} n \ln(\sqrt{2n}) \|A\|_F^2$.
3. **For each** index $(i, j) \in [M] \times [N]$, set
$$p_{ij} = \begin{cases} s\frac{B_{ij}^2}{\|B\|_F^2} & \text{if } sB_{ij}^2 \leq \|B\|_F^2 \\ 1 & \text{if } sB_{ij}^2 > \|B\|_F^2 . \end{cases}$$
4. **Define** the sparse random matrix A^{sp} with entries
$$A_{ij}^{sp} = \begin{cases} \frac{B_{ij}}{p_{ij}} & \text{with probability } p_{ij} \\ 0 & \text{with probability } 1 - p_{ij}. \end{cases}$$

Output: $A^{sp} \in \mathbb{R}^{M\times N}$ satisfying $\|A - A^{sp}\|_2 < \delta$.

We now describe the algorithm at a high-level. First, in the same manner as **[18]**, we zero out the "small" entries of A. Then, we deterministically preserve the "large" entries, because we need to ensure that our sampling probabilities remain in $[0, 1]$. Lastly and perhaps most notably, we independently randomly sample the majority of entries of A (i.e., without replacement), with sampling probabilities proportional to the square of the entry. The entries that are preserved are strengthened by the reciprocal of the sampling probability.

Theorem 4.1. *Let $\delta > 0$, $A \in \mathbb{R}^{M\times N}$ be a matrix, and let $n = \max\{M, N\}$. Define $s = \frac{128}{3\delta^2} n \ln(\sqrt{2n}) \|A\|_F^2$. The random matrix A^{sp} constructed in Algorithm 3 satisfies*

$$\|A - A^{sp}\|_2 \leq \delta$$

with high probability. Furthermore, A^{sp} has $s = O(n \log n \|A\|_F^2)$ expected number of non-zero entries.

The proof of Theorem 4.1 is fairly involved, so we postpone it until Section 4.5. We pause here to note that Theorem 4.1 provides an absolute error bound, while noise-prune guarantees a relative error bound

$$\|A - A^{sp}\|_2 \leq \delta \|A\|_2 \qquad \text{(See Eq. 3.42)}.$$

Actually, because the matrix A can be fixed before choosing the error tolerance δ, we can simply choose δ as a function of $\|A\|_2$, so both Theorem 4.1 and noise-prune can be formulated with either relative or absolute errors via a simple scaling argument. Below we present the relative error version of Theorem 4.1.

Corollary 4.2. *Let $\tilde{\delta} > 0$, $A \in \mathbb{R}^{M\times N}$ be a matrix, and let $n = \max\{M, N\}$. Define $s = \frac{128}{3\tilde{\delta}^2} n \ln(\sqrt{2n}) \frac{\|A\|_F^2}{\|A\|_2^2}$. The random matrix A^{sp} constructed in Algorithm 3 satisfies*

$$\|A - A^{sp}\|_2 \leq \tilde{\delta} \|A\|_2$$

with high probability. Furthermore, A^{sp} has $s = O(n \log n \frac{\|A\|_F^2}{\|A\|_2^2})$ expected number of non-zero entries.

Proof. Let $\tilde{\delta} > 0$ be fixed. Define $\delta = \tilde{\delta} \|A\|_2$ and

$$s = \frac{128}{3\delta^2} n \ln(\sqrt{2n}) \|A\|_F^2 = \frac{128}{3\tilde{\delta}^2} n \ln(\sqrt{2n}) \frac{\|A\|_F^2}{\|A\|_2^2}.$$

By Theorem 4.1, we have that

$$\|A - A^{sp}\|_2 \leq \delta = \tilde{\delta}\,\|A\|_2\,,$$

with high probability. □

The quantity $\frac{\|A\|_F^2}{\|A\|_2^2}$ is sometimes called the *stable rank* of A, because the squared Frobenius norm is the sum of squared singular values of A while the squared 2-norm is simply the largest squared singular value of A. This stability is in the numerical sense. As an example, a stable rank of 3 is implied when approximately 3 singular values are on the order of magnitude of the largest singular value.

Before we prove Theorem 4.1, we describe the powerful rectangular matrix inequality that provides us the machinery to do so.

4.4.1. The Rectangular Matrix Bernstein Inequality. Like many other concentration of measure inequalities, the matrix Bernstein inequality was originally proven for symmetric (hence square) matrices by Gross et al. in 2010 [**29**]. This lack of generality leaves those working with neural networks wanting, as weight matrices are rarely symmetric, and are rectangular in general due to the non-uniformity of layer sizes. Fortunately, Recht and Tropp independently extended the inequality to rectangular matrices soon afterward in 2011, cleverly importing the technique of *self-adjoint dilation* from operator theory [**70**, **90**]. This formulation is the one that is useful for our purposes. We state the rectangular Bernstein inequality below.

Theorem 4.3. *Let $X_1, \ldots, X_L$ be independent zero-mean random matrices of dimension $M \times N$, such that $\|X_k\|_2 \leq R$ almost surely for all $k \in [L]$. For any $t \geq 0$, we have*

$$P\left[\left\|\sum_{k=1}^{L} X_k\right\| \geq t\right] \leq (M+N)\exp\left(-\frac{t^2/2}{\sigma^2 + Rt/3}\right),$$

where $\sigma^2 = \max\left\{\left\|\sum_{k=1}^{L} X_k X_k^T\right\|_2, \left\|\sum_{k=1}^{L} X_k^T X_k\right\|_2\right\}$.

We will describe precisely how we utilize this inequality in the proof of Theorem 4.1.

4.5. Proof of Theorem 4.1

Because of its reliance on the rectangular matrix Bernstein inequality and many technical lemmas, the proof of Theorem 4.1 will have the entirety of this section devoted to it. For brevity in lemma statements, we always refer to a fixed $\delta > 0$, $M, N \in \mathbb{N}$, and $A \in \mathbb{R}^{M \times N}$. We follow a modification of the overall proof strategy used in **[18]**. Our algorithm samples matrix entries without replacement, so the distribution of each random matrix used in Theorem 4.3 is Bernoulli in one entry (and identically 0 in all other entries), rather than uniform over all entries. This algorithmic distinction leaves the equivalent of Lemma 4.4 unchanged, but requires the manipulation of different bounds on R and σ^2 in Lemmas 4.5 and 4.6. The theoretical guarantees on expected non-zero entries remains unchanged.

The structure of the overall argument will bound $\|A - A^{sp}\|_2$ using a standard $\delta/2$-argument:

$$\begin{aligned} \|A - A^{sp}\|_2 &= \|A - B + B - A^{sp}\|_2 \\ &\leq \|A - B\|_2 + \|A^{sp} - B\|_2 \\ &\leq \frac{\delta}{2} + \frac{\delta}{2} = \delta. \end{aligned}$$

Lemma 4.4 will provide the bound on $\|A - B\|_2$, while Theorem 4.3 will provide the high-probability bound on $\|A^{sp} - B\|_2$ through Lemmas 4.5 and 4.6. Following this, we will compute the expected number of non-zero entries in A^{sp}.

4.5.1. Bounding $\|A - B\|_2$. The construction of B simply deterministically prunes away sufficiently small entries. This subtle step is necessary to find an upper bound on the reciprocal of our weights, which is utilized in the proof of Lemma 4.5.

Lemma 4.4. *Define the matrix B by*

$$B_{ij} = \begin{cases} A_{ij} & \text{if } |A_{ij}| \geq \frac{\delta}{2n} \\ 0 & \text{if } |A_{ij}| < \frac{\delta}{2n}. \end{cases}$$

Then, we have

$$\|A - B\|_2 \leq \frac{\delta}{2}$$

PROOF. Since

$$(A-B)_{ij} = \begin{cases} 0 & \text{if } |A_{ij}| \geq \frac{\delta}{2n} \\ A_{ij} & \text{if } |A_{ij}| < \frac{\delta}{2n}, \end{cases}$$

we can see that $(A-B)_{ij} \leq \frac{\delta}{2n}$ for each (i,j).

Recalling that the 2-norm is bounded by the Frobenius norm, we have

$$\|A-B\|_2^2 \leq \|A-B\|_F^2 = \sum_{i=1}^{M}\sum_{j=1}^{N}(A-B)_{ij}^2 \leq \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{\delta^2}{4n^2} = \frac{\delta^2 MN}{4n^2} \leq \frac{\delta^2}{4},$$

so

$$\|A-B\|_2 \leq \frac{\delta}{2}.$$

□

4.5.2. Bounding $\|A^{sp}-B\|_2$ with the matrix Bernstein inequality. We will apply Theorem 4.3 to the matrices $\{X^{(i,j)}\}$ indexed by $(i,j) \in [M] \times [N]$, where each $X^{(i,j)}$ has all entries equal to 0 except the (i,j)th entry, given by

$$X_{ij}^{(i,j)} = \begin{cases} \frac{B_{ij}}{p_{ij}} - B_{ij} & \text{with probability } p_{ij} \\ -B_{ij} & \text{with probability } 1-p_{ij}. \end{cases}$$

Note that $\mathbb{E}[X^{(i,j)}] = 0$, since $\mathbb{E}[X_{ij}^{(i,j)}] = \left(\frac{B_{ij}}{p_{ij}} - B_{ij}\right) p_{ij} + (-B_{ij})(1-p_{ij}) = 0$, and all other entries are identically 0. Note also that $\sum_{(i,j)} X^{(i,j)} = A^{sp} - B$, with A^{sp} defined as in Algorithm 3:

$$A_{ij}^{sp} = \begin{cases} \frac{B_{ij}}{p_{ij}} & \text{with probability } p_{ij} \\ 0 & \text{with probability } 1-p_{ij}. \end{cases}$$

Hence, taking $t = \delta/2$, Theorem 4.3 describes the inequality

$$P\left[\|A^{sp}-B\| \geq \frac{\delta}{2}\right] \leq (M+N)\exp\left(-\frac{\delta^2/8}{\sigma^2 + R\delta/6}\right). \tag{4.2}$$

If we can guarantee that the right-hand-side of Eq. 4.2 is $O(1/n)$, then we have

$$\|A^{sp}-B\| < \frac{\delta}{2}$$

with high probability, as desired. Thus, the next two subsections focus on bounding R and σ^2 respectively.

4.5.3. Bounding R.

Lemma 4.5. *For each $(i,j) \in [M] \times [N]$, we have*

$$\left\| X^{(i,j)} \right\|_2 \leq \frac{4n}{\delta s} \|B\|_F^2 .$$

Proof. Notice that $\left\|X^{(i,j)}\right\|_2 \leq \left\|X^{(i,j)}\right\|_F \leq \max\left\{|\frac{B_{ij}}{p_{ij}} - B_{ij}|, |B_{ij}|\right\} \leq |\frac{B_{ij}}{p_{ij}}| + |B_{ij}|$. Then, we have

$$\begin{aligned}
|\frac{B_{ij}}{p_{ij}}| + |B_{ij}| &= \frac{\|B\|_F^2}{s|B_{ij}|} + |B_{ij}| && \text{by definition of } p_{ij} \\
&\leq \frac{\|B\|_F^2}{s|B_{ij}|} + \|B\|_F && \text{by definition of } \|\cdot\|_F \\
&\leq \frac{2n}{\delta s} \|B\|_F^2 + \|B\|_F && \text{since } |B_{ij}| \geq \frac{\delta}{2n}.
\end{aligned}$$

So, it remains to show that $\|B\|_F \leq \frac{2n}{\delta s} \|B\|_F^2$.

To see that this must be true, suppose for contradiction that $\|B\|_F > \frac{2n}{\delta s} \|B\|_F^2$. Rearranging this inequality reveals that

$$\|B\|_F < \frac{\delta s}{2n},$$

which is a contradiction as long as B has at least s non-zero entries, since each element of B is at least $\delta/2n$ in magnitude by construction (note: if B has less than s entries, we do not need to prune it, as it is already as sparse as Theorem 4.1 guarantees it's sparsification would be). Putting this together with our string of inequalities, we have

$$\begin{aligned}
\left\| X^{(i,j)} \right\|_2 &\leq \frac{2n}{\delta s} \|B\|_F^2 + \|B\|_F \\
&\leq \frac{2n}{\delta s} \|B\|_F^2 + \frac{2n}{\delta s} \|B\|_F^2 \\
&= \frac{4n}{\delta s} \|B\|_F^2 ,
\end{aligned}$$

as desired. □

4.5.4. Bounding σ^2.

Lemma 4.6. *Given* $\sigma^2 = \max\left\{\left\|\sum_{(i,j)} \mathbb{E}\left[X^{(i,j)}(X^{(i,j)})^T\right]\right\|_2, \left\|\sum_{(i,j)} \mathbb{E}\left[(X^{(i,j)})^T X^{(i,j)}\right]\right\|_2\right\}$, *we have*

$$\sigma^2 \leq \frac{2n}{s}\|B\|_F^2.$$

Proof. We first focus on bounding $\left\|\sum_{(i,j)} \mathbb{E}\left[X^{(i,j)}(X^{(i,j)})^T\right]\right\|_2$ (and will later show that $\left\|\sum_{(i,j)} \mathbb{E}\left[(X^{(i,j)})^T X^{(i,j)}\right]\right\|_2$ is bounded by the same quantity). For each (i,j), we have

$$\begin{aligned}
\mathbb{E}\left[X^{(i,j)}(X^{(i,j)})^T\right] &= p_{ij}(B_{ij}(\frac{1}{p_{ij}}-1)e_ie_j^T)(B_{ij}(\frac{1}{p_{ij}}-1)e_je_i^T) \\
&\quad + (1-p_{ij})(-B_{ij}e_ie_j^T)(-B_{ij}e_je_i^T) \\
&= B_{ij}^2 p_{ij}(\frac{1}{p_{ij}^2}-\frac{2}{p_{ij}}+1)e_ie_i^T + B_{ij}^2(1-p_{ij})e_ie_i^T \\
&= B_{ij}^2(\frac{1}{p_{ij}}-1)e_ie_i^T.
\end{aligned}$$

In other words, each term $\mathbb{E}\left[X^{(i,j)}(X^{(i,j)})^T\right]$ contributes only to the diagonal elements of the resulting summation over (i,j). Indeed, if we define $C = \sum_{(i,j)} \mathbb{E}\left[X^{(i,j)}(X^{(i,j)})^T\right]$, we see that $C_{ij} = 0$ when $i \neq j$, and

$$C_{ii} = \sum_{j=1}^{N} B_{ij}^2(\frac{1}{p_{ij}}-1).$$

As a result of C being a diagonal matrix, $\|C\|_2$ is simply the maximum of its diagonal elements, so we can bound it as follows:

$$\begin{aligned}
\|C\|_2 &= \max_{1\le i\le M}\left(\sum_{j=1}^{N} B_{ij}^2(\frac{1}{p_{ij}}-1)\right)\\
&\le \max_{1\le i\le M}\left(\sum_{j=1}^{N} \frac{B_{ij}^2}{p_{ij}}+B_{ij}^2\right)\\
&\le \max_{1\le i\le M}\left(\sum_{j=1}^{N} \frac{B_{ij}^2}{p_{ij}}+\frac{B_{ij}^2}{p_{ij}}\right) && \text{since } 0<p_{ij}<1\\
&= \max_{1\le i\le M}\left(\sum_{j=1}^{N} \frac{2B_{ij}^2}{p_{ij}}\right)\\
&\le \max_{1\le i\le M}\left(\sum_{j=1}^{N} \frac{2\|B\|_F^2}{s}\right)\\
&= \frac{2N}{s}\|B\|_F^2 \le \frac{2n}{s}\|B\|_F^2 .
\end{aligned}$$

Following a very similar argument, if we define $D=\sum_{(i,j)}\mathbb{E}\left[(X^{(i,j)})^T X^{(i,j)}\right]$, we find that

$$\mathbb{E}\left[(X^{(i,j)})^T X^{(i,j)}\right] = B_{ij}^2(\frac{1}{p_{ij}}-1)e_j e_j^T,$$

which allows us to see that D is a diagonal matrix with entries

$$D_{jj}=\sum_{i=1}^{M} B_{ij}^2(\frac{1}{p_{ij}}-1).$$

As such, we find that

$$\|D\|_2 \le \frac{2M}{s}\|B\|_F^2 \le \frac{2n}{s}\|B\|_F^2 .$$

In any case, we find that

$$\sigma^2 = \max\{\|C\|_F^2, \|D\|_F^2\} \le \frac{2n}{s}\|B\|_F^2 .$$

□

4.5.5. Tying together R, σ^2, and Eq. 4.2. Now that we have found an expression for R and a bound for σ^2, we can continue to inspect Eq. 4.2:

$$\begin{aligned}
P\left[\|A^{sp}-B\| \geq \frac{\delta}{2}\right] &\leq (M+N)\exp\left(-\frac{\delta^2/8}{\sigma^2+R\delta/6}\right) \\
&= (M+N)\exp\left(-\frac{\delta^2/8}{\sigma^2+\frac{4n}{\delta s}\|B\|_F^2\frac{\delta}{6}}\right) && \text{since } R=\frac{4n}{\delta s}\|B\|_F^2 \\
&\leq 2n\exp\left(-\frac{\delta^2/8}{\frac{2n}{s}\|B\|_F^2+\frac{2n}{3s}\|B\|_F^2}\right) && \text{since } \sigma^2 \leq \frac{2n}{s}\|B\|_F^2 \\
&\leq 2n\exp\left(-s\frac{3\delta^2}{64n\|B\|_F^2}\right) \\
&\leq 2n\exp\left(-\ln(2n^2)\frac{\|A\|_F^2}{\|B\|_F^2}\right) && \text{since } s=\frac{64}{3\delta^2}n\ln(2n^2)\|A\|_F^2 \\
&\leq 2n\exp\left(-\ln(2n^2)\right) && \text{since } \|B\|_F \leq \|A\|_F \text{ by construction} \\
&= \frac{1}{n}.
\end{aligned}$$

Hence, with high probability, we have $\|A^{sp}-B\|_2 < \frac{\delta}{2}$. This completes the proof that $\|A-A^{sp}\|_2 \leq \delta$. It is perhaps worth noting that the choice of s is somewhat arbitrary, in the sense that we could choose a larger s to arrive at a lower failure probability than $1/n$ (and similarly, we could choose a smaller s to arrive at a higher failure probability). This is noteworthy because s governs the sparsity of the pruned matrix, as we will see in the next subsection.

4.5.6. Bounding the number of nonzero entries of A^{sp}. Recall that A^{sp} is given by

$$A_{ij}^{sp} = \begin{cases} \frac{B_{ij}}{p_{ij}} & \text{with probability } p_{ij} \\ 0 & \text{with probability } 1-p_{ij} \end{cases}$$

with

$$p_{ij} = \begin{cases} s\frac{B_{ij}^2}{\|B\|_F^2} & \text{if } sB_{ij}^2 \leq \|B\|_F^2 \\ 1 & \text{if } sB_{ij}^2 > \|B\|_F^2 . \end{cases}$$

As such, note that the expected number of nonzero entries of A^{sp} is simply $\sum_{(i,j)\in[M]\times[N]} p_{ij}$. We will partition $[M]\times[N]$ by considering the p_{ij} that are smaller than 1 and the p_{ij} that are identically 1.

That is, let $S = \{(i,j) \in [M] \times [N] : sB_{ij}^2 \leq \|B\|_F^2\}$ so that $S^c = \{(i,j) \in [M] \times [N] : sB_{ij}^2 > \|B\|_F^2\}$. With this notation in mind, we can readily see that

$$\begin{aligned}
\sum_{(i,j)\in[M]\times[N]} p_{ij} &= \sum_{(i,j)\in S} p_{ij} + \sum_{(i,j)\in S^c} p_{ij} \\
&= \sum_{(i,j)\in S} s\frac{B_{ij}^2}{\|B\|_F^2} + \sum_{(i,j)\in S^c} 1 \\
&\leq \sum_{(i,j)\in S} s\frac{B_{ij}^2}{\|B\|_F^2} + \sum_{(i,j)\in S^c} s\frac{B_{ij}^2}{\|B\|_F^2} \qquad \text{since } sB_{ij}^2 > \|B\|_F^2 \text{ when } (i,j) \in S^c \\
&= \sum_{(i,j)\in[M]\times[N]} \frac{sB_{ij}^2}{\|B\|_F^2} \\
&= \frac{s}{\|B\|_F^2} \sum_{(i,j)\in[M]\times[N]} B_{ij}^2 \\
&= s \qquad \text{since } \sum_{(i,j)\in[M]\times[N]} B_{ij}^2 = \|B\|_F^2 .
\end{aligned}$$

Hence, the expected number of non-zero entries of A^{sp} is at most s, which completes the proof.

4.6. Error Bounds for Pruned Feed-Forward Neural Networks

Now that we have the ability to create sparse rectangular weight matrices with similar spectral properties to their original dense counterparts, we consider the error between the output of a feed-forward network comprised of pruned weight matrices and that of the original, dense network. In fact, we construct error bounds that apply at any activity layer of a deep network, allowing us to easily analogize our result to discrete-time recurrent neural networks. Throughout Sections 4.6 and 4.7, we follow the convention that $\|\cdot\| = \|\cdot\|_2$ unless otherwise stated.

Theorem 4.7. *Let $G(\boldsymbol{x}, \boldsymbol{\Theta})$ be a feed-forward neural network with a contraction activation function $\boldsymbol{g}(\cdot)$ (i.e., $\boldsymbol{g}$ has Lipschitz constant $0 \leq k \leq 1$) applied after each layer with input $\boldsymbol{x} \in \mathbb{R}^{N_0}$ and parameters*

$$\boldsymbol{\Theta} = (W^{(1)}, W^{(2)}, \ldots, W^{(n)}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}, \ldots, \boldsymbol{b}^{(n)}),$$

where $W^{(\ell)} \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ *and* $b^{(\ell)} \in \mathbb{R}^{N_\ell}$*. That is, the activity at layer* ℓ *is given by*

$$\boldsymbol{x}^{(\ell)} = \boldsymbol{g}(W^{(\ell)}\boldsymbol{x}^{(\ell-1)} + \boldsymbol{b}^{(\ell)}) \quad \text{for } \ell = 1, \ldots, n,$$

so that $\boldsymbol{x}^{(n)} = G(\boldsymbol{x}^{(0)}, \boldsymbol{\Theta}) \in \mathbb{R}^{N_n}$*. Furthermore, assume* $G(\boldsymbol{x}, \boldsymbol{\Theta}^{sp})$ *is a layer-wise pruned neural network with parameters*

$$\boldsymbol{\Theta}^{sp} = (W^{(1,sp)}, W^{(2,sp)}, \ldots, W^{(n,sp)}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}, \ldots, \boldsymbol{b}^{(n)})$$

where $\left\|W^{(\ell)} - W^{(\ell,sp)}\right\| \leq \delta_\ell$ *for each* $\ell = 1, \ldots, n$*. The activity at layer* ℓ *is given by*

$$\boldsymbol{x}^{(\ell,sp)} = \boldsymbol{g}(W^{(\ell,sp)}\boldsymbol{x}^{(\ell-1,sp)} + \boldsymbol{b}^{(\ell)}) \quad \text{for } \ell = 1, \ldots, n.$$

For any layer ℓ *such that* $1 \leq \ell \leq n$*, if* $\boldsymbol{x}^{(0)} = \boldsymbol{x}^{(0,sp)}$*, we have the error bound*

$$\left\|\boldsymbol{x}^{(\ell)} - \boldsymbol{x}^{(\ell,sp)}\right\| \leq \sum_{j=0}^{\ell-1} k^{j+1}\delta_{\ell-j} \left\|\boldsymbol{x}^{(\ell-1-j,sp)}\right\| \cdot \prod_{i=0}^{j-1} \left\|W^{(\ell-i)}\right\|. \tag{4.3}$$

PROOF. Let $1 \leq \ell \leq n$ be fixed. More generally, we claim that

$$\left\|\boldsymbol{x}^{(\ell)} - \boldsymbol{x}^{(\ell,sp)}\right\| \leq \left\|\boldsymbol{x}^{(\ell-m)} - \boldsymbol{x}^{(\ell-m,sp)}\right\| k^m \cdot \prod_{j=0}^{m-1} \left\|W^{(\ell-j)}\right\| + \sum_{j=0}^{m-1} k^{j+1}\delta_{\ell-j} \left\|\boldsymbol{x}^{(\ell-1-j,sp)}\right\| \cdot \prod_{i=0}^{j-1} \left\|W^{(\ell-i)}\right\| \tag{4.4}$$

for all $m = 1, \ldots, \ell$ (note: in the case $m = 1$, we follow the convention that an empty product is equal to 1). Once we show Eq. (4.4), taking $m = \ell$ achieves the result stated in the theorem, since $\boldsymbol{x}^{(0)} = \boldsymbol{x}^{(0,sp)}$ by assumption. We show the base case $m = 1$ below.

$$\begin{aligned}
\left\|\boldsymbol{x}^{(\ell)} - \boldsymbol{x}^{(\ell,sp)}\right\| &= \left\|\boldsymbol{g}(W^{(\ell)}\boldsymbol{x}^{(\ell-1)} + \boldsymbol{b}^{(\ell)}) - \boldsymbol{g}(W^{(\ell,sp)}\boldsymbol{x}^{(\ell-1,sp)} + \boldsymbol{b}^{(\ell)})\right\| \\
&\leq k \left\|W^{(\ell)}\boldsymbol{x}^{(\ell-1)} + \boldsymbol{b}^{(\ell)} - (W^{(\ell,sp)}\boldsymbol{x}^{(\ell-1,sp)} + \boldsymbol{b}^{(\ell)})\right\| \\
&\leq k \left\|W^{(\ell)}\boldsymbol{x}^{(\ell-1)} - W^{(\ell,sp)}\boldsymbol{x}^{(\ell-1,sp)}\right\| \\
&\leq k \left\|W^{(\ell)}\boldsymbol{x}^{(\ell-1)} - (W^{(\ell)} - W^{(\ell)} + W^{(\ell,sp)})\boldsymbol{x}^{(\ell-1,sp)}\right\| \\
&\leq k \left(\left\|W^{(\ell)}\right\| \left\|\boldsymbol{x}^{(\ell-1)} - \boldsymbol{x}^{(\ell-1,sp)}\right\| + \left\|W^{(\ell)} - W^{(\ell,sp)}\right\| \left\|x^{(\ell-1,sp)}\right\|\right) \\
&\leq k \left(\left\|W^{(\ell)}\right\| \left\|\boldsymbol{x}^{(\ell-1)} - \boldsymbol{x}^{(\ell-1,sp)}\right\| + \delta_\ell \left\|x^{(\ell-1,sp)}\right\|\right)
\end{aligned}$$

Now suppose Eq. (4.4) holds for $m-1 \in \{1,\dots,\ell-1\}$. We will show that it holds for m as well. By the inductive hypothesis, we have

$$\begin{aligned}\left\|\boldsymbol{x}^{(\ell)}-\boldsymbol{x}^{(\ell,sp)}\right\| &\leq \left\|\boldsymbol{x}^{(\ell-m+1)}-\boldsymbol{x}^{(\ell-m+1,sp)}\right\| k^{m-1}\cdot\prod_{j=0}^{m-2}\left\|W^{(\ell-j)}\right\| \\ &\quad+\sum_{j=0}^{m-2}k^{j+1}\delta_{\ell-j}\left\|\boldsymbol{x}^{(\ell-1-j,sp)}\right\|\cdot\prod_{i=0}^{j-1}\left\|W^{(\ell-i)}\right\| \\ =\left\|\boldsymbol{g}(W^{(\ell-m+1)}\boldsymbol{x}^{(\ell-m)}+\boldsymbol{b}^{(\ell-m+1)})-\boldsymbol{g}(W^{(\ell-m+1,sp)}\boldsymbol{x}^{(\ell-m,sp)}+\boldsymbol{b}^{(\ell-m+1)})\right\| k^{m-1}\cdot\prod_{j=0}^{m-2}\left\|W^{(\ell-j)}\right\| \\ &\quad+\sum_{j=0}^{m-2}k^{j+1}\delta_{\ell-j}\left\|\boldsymbol{x}^{(\ell-1-j,sp)}\right\|\cdot\prod_{i=0}^{j-1}\left\|W^{(\ell-i)}\right\|.\end{aligned}$$

Now, consider the expression

$$\left\|\boldsymbol{g}(W^{(\ell-m+1)}\boldsymbol{x}^{(\ell-m)}+\boldsymbol{b}^{(\ell-m+1)})-\boldsymbol{g}(W^{(\ell-m+1,sp)}\boldsymbol{x}^{(\ell-m,sp)}+\boldsymbol{b}^{(\ell-m+1)})\right\|,$$

which is bounded above by

$$\begin{aligned}&\leq k\left\|W^{(\ell-m+1)}\boldsymbol{x}^{(\ell-m)}-W^{(\ell-m+1,sp)}\boldsymbol{x}^{(\ell-m,sp)}\right\| \\ &= k\left\|W^{(\ell-m+1)}\boldsymbol{x}^{(\ell-m)}-(W^{(\ell-m+1)}-W^{(\ell-m+1)}+W^{(\ell-m+1,sp)})\boldsymbol{x}^{(\ell-m,sp)}\right\| \\ &\leq k\left(\left\|W^{(\ell-m+1)}\right\|\left\|\boldsymbol{x}^{(\ell-m)}-\boldsymbol{x}^{(\ell-m,sp)}\right\|+\left\|W^{(\ell-m+1)}-W^{(\ell-m+1,sp)}\right\|\left\|\boldsymbol{x}^{(\ell-m,sp)}\right\|\right) \\ &= k\left(\left\|W^{(\ell-m+1)}\right\|\left\|\boldsymbol{x}^{(\ell-m)}-\boldsymbol{x}^{(\ell-m,sp)}\right\|+\delta_{\ell-m+1}\left\|\boldsymbol{x}^{(\ell-m,sp)}\right\|\right).\end{aligned}$$

Thus, we have

$$\begin{aligned}\left\|\boldsymbol{x}^{(\ell)}-\boldsymbol{x}^{(\ell,sp)}\right\| &\leq k\left(\left\|W^{(\ell-m+1)}\right\|\left\|\boldsymbol{x}^{(\ell-m)}-\boldsymbol{x}^{(\ell-m,sp)}\right\|+\delta_{\ell-m+1}\left\|\boldsymbol{x}^{(\ell-m,sp)}\right\|\right)k^{m-1}\cdot\prod_{j=0}^{m-2}\left\|W^{(\ell-j)}\right\| \\ &\quad+\sum_{j=0}^{m-2}k^{j+1}\delta_{\ell-j}\left\|\boldsymbol{x}^{(\ell-1-j,sp)}\right\|\cdot\prod_{i=0}^{j-1}\left\|W^{(\ell-i)}\right\| \\ =\left\|\boldsymbol{x}^{(\ell-m)}-\boldsymbol{x}^{(\ell-m,sp)}\right\| k^{m}\cdot\prod_{j=0}^{m-1}\left\|W^{(\ell-j)}\right\|+\sum_{j=0}^{m-1}k^{j+1}\delta_{\ell-j}\left\|\boldsymbol{x}^{(\ell-1-j,sp)}\right\|\cdot\prod_{i=0}^{j-1}\left\|W^{(\ell-i)}\right\|,\end{aligned}$$

precisely as desired, proving the result by induction. □

4.7. Corollaries of Theorem 4.7

Theorem 4.7 is presented in the most general sense, with no concrete activation function and no restriction on the weight matrices. We now highlight how the error bound Eq. 4.1 is affected by choosing bounded Lipschitz activation functions with $0 \leq k \leq 1$, such as the standard sigmoid. We also highlight the simplicity of the error bound when weight matrices are unitary, and that the bound applies to discrete-time recurrent neural networks as well; this point is especially salient, as networks of this form are now frequently called unitary recurrent neural networks (URNNs), and they have become quite popular [**2**, **20**, **40**, **52**, **97**]. URNNs have been shown to be as expressive as general RNNs [**20**, **97**], while largely circumventing the exploding/vanishing gradient problem that traditionally plagues recurrent networks [**2**]. In essence, the unitary limitation on weight matrix norm is not nearly as limiting as one might believe, and they make our results far more appealing due to our error bounds being functions of the matrix norms.

Corollary 4.8. *Let $G(\boldsymbol{x}, \boldsymbol{\Theta})$ and $G(\boldsymbol{x}, \boldsymbol{\Theta}^{sp})$ be as defined in Theorem 4.1. Furthermore, assume that the sigmoid $\boldsymbol{g}(\cdot)$ is given by the entrywise logistic function $g : \mathbb{R} \to \mathbb{R}$ with $g(x) = \frac{e^x}{e^x+1}$.*

For any layer ℓ such that $1 \leq \ell \leq n$, if $\boldsymbol{x}^{(0)} = \boldsymbol{x}^{(0,sp)}$ with $\left\|\boldsymbol{x}^{(0)}\right\| \leq 1$, we have the error bound

$$\left\|\boldsymbol{x}^{(\ell)} - \boldsymbol{x}^{(\ell,sp)}\right\| \leq \sum_{j=0}^{\ell-1} \delta_{\ell-j} \cdot \prod_{i=0}^{j-1} \left\|W^{(\ell-i)}\right\|. \tag{4.5}$$

Moreover, if each of the original weight matrices are unitary, we have the bound

$$\left\|\boldsymbol{x}^{(\ell)} - \boldsymbol{x}^{(\ell,sp)}\right\| \leq \sum_{j=1}^{\ell} \delta_j. \tag{4.6}$$

Proof. We begin by noting that $\boldsymbol{g}(\cdot)$ has Lipschitz constant $k = 1$, because the one-dimensional $g(x)$ has Lipschitz constant $k = 1$ (see Proposition 4.12 in the Appendix for a proof of this). Thus, in order to convert Eq. 4.3 to Eq. 4.5, it suffices to observe the bound of each layer's activity of the sparsified network:

$$\left\|\boldsymbol{x}^{(j,sp)}\right\| \leq 1 \quad \text{for } j = 0, \ldots, \ell - 1$$

Note that $j = 0$ holds by assumption. Now, recall that

$$\left\|\boldsymbol{x}^{(j,sp)}\right\| = \left\|\boldsymbol{g}(W^{(j,sp)}\boldsymbol{x}^{(j-1,sp)} + \boldsymbol{b}^{(j)})\right\| \leq 1 \quad \text{for } j = 1, \ldots, \ell - 1,$$

simply because the sigmoid function is always bounded by 1. So we arrive at the desired inequality

$$\left\|\boldsymbol{x}^{(\ell)} - \boldsymbol{x}^{(\ell,sp)}\right\| \leq \sum_{j=0}^{\ell-1} \delta_{\ell-j} \cdot \prod_{i=0}^{j-1} \left\|W^{(\ell-i)}\right\|.$$

Now, if each of the weight matrices are unitary (i.e., $\left\|W^{(\ell-i)}\right\| = 1$ for $i = 0, \ldots, j-1$), we immediately arrive at

$$\left\|\boldsymbol{x}^{(\ell)} - \boldsymbol{x}^{(\ell,sp)}\right\| \leq \sum_{j=0}^{\ell-1} \delta_{\ell-j} = \sum_{j=1}^{\ell} \delta_j,$$

completing the proof. □

That is, given a desired degree of approximation $\epsilon > 0$, if we want

$$\left\|\boldsymbol{x}^{(\ell)} - \boldsymbol{x}^{(\ell,sp)}\right\| \leq \epsilon,$$

we can simply choose $\delta_j = \frac{\epsilon}{\ell}$ for $j = 1, \ldots, \ell$, as long as our weight matrices are unitary. In that case, each matrix in our network will have $O(\ell^2 N_j \log N_j \epsilon^{-2})$ non-zero entries, which means that the total number of parameters in the network is $O(\ell^3 N_{max} \log N_{max} \epsilon^{-2})$, where $N_{max} = \max_j N_j$. We note that a similar result can be shown for any contraction $\boldsymbol{g}$ that is bounded, but chose the logistic function because it is bounded by 1 for simplicity. We now show that Theorem 4.7 can be ported into the recurrent neural network setting in the next corollary.

Corollary 4.9. *Let $\boldsymbol{x}^{(t+1)} = \boldsymbol{g}(W\boldsymbol{x}^{(t)} + \boldsymbol{b}^{(t)})$ be a discrete-time recurrent neural network with weight matrix $W \in \mathbb{R}^{n\times n}$, bias vectors (i.e., "input") $\boldsymbol{b}^{(t)} \in \mathbb{R}^n$, and activation function $\boldsymbol{g}(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ with Lipschitz constant $0 \leq k \leq 1$. Furthermore, assume $x^{(t+1,sp)} = \boldsymbol{g}(W^{sp}\boldsymbol{x}^{(t+1,sp)} + \boldsymbol{b}^{(t)})$ is a pruned recurrent neural network satisfying $\|W - W^{sp}\| \leq \delta$. For any time-step $t \geq 1$, if $\boldsymbol{x}^{(0)} = \boldsymbol{x}^{(0,sp)}$, we have the error bound*

$$\left\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t,sp)}\right\| \leq \delta \sum_{j=0}^{t-1} k^{j+1} \left\|\boldsymbol{x}^{(t-1-j,sp)}\right\| \|W\|^j.$$

Proof. Let $t \geq 1$ and define the feed-forward network $G(\boldsymbol{x}, \boldsymbol{\Theta})$ with t layers, where each $W^{(\ell)} = W$ and $b^{(\ell)}$ is indexed by $\ell = 1, \ldots, t$. Next, construct its sparse counterpart $G(\boldsymbol{x}, \boldsymbol{\Theta}^{sp})$ with each $W^{(\ell,sp)} = W$. Note that

$$\left\|W^{(\ell)} - W^{(\ell,sp)}\right\| = \|W - W^{sp}\| \leq \delta.$$

Then, applying Theorem 4.7 with each $\delta_{\ell-j} = \delta$ and $\left\|W^{(\ell-i)}\right\| = \|W\|$, we have

$$\left\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t,sp)}\right\| \leq \delta \sum_{j=0}^{t-1} k^{j+1} \left\|\boldsymbol{x}^{(t-1-j,sp)}\right\| \|W\|^j ,$$

as desired. □

Corollary 4.9, as presently stated, bounds the norm of the difference of activities at the t-th timestep, and may be useful for vector-to-vector recurrent network structures that are only interested in such outputs. For example, Hopfield networks tend to only concern themselves with the steady-state output, a single vector, typically encoding a single image (sometimes called a pattern or memory).

Some recurrent networks are trained to perform on tasks that consider their "input" to be the sequence of bias vectors (as noted in the corollary statement), and these tasks are formulated in such a way that they tend to consider their "output" to be the sequence of neural activity given by $X = (\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(t)})$, which can be thought of as a matrix with iT-th entry $X_{iT} = x_i^{(T)}$. Sequence-to-sequence architectures of this sort are often used to solve problems in speech recognition and natural language processing and are an extremely active area of research **[24, 30, 39, 55, 87]**. We note that we can extend Corollary 4.9 to bound the error of outputs in the sequence-to-sequence setting in Frobenius norm.

Corollary 4.10. *Let $\boldsymbol{x}^{(t+1)} = \boldsymbol{g}(W\boldsymbol{x}^{(t)} + \boldsymbol{b}^{(t)})$ be a discrete-time recurrent neural network with weight matrix $W \in \mathbb{R}^{n\times n}$, bias vectors (i.e., "input") $\boldsymbol{b}^{(t)} \in \mathbb{R}^n$, and activation function $\boldsymbol{g}(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ with Lipschitz constant $0 \leq k \leq 1$. Furthermore, assume $x^{(t+1,sp)} = \boldsymbol{g}(W^{sp}\boldsymbol{x}^{(t+1,sp)} + \boldsymbol{b}^{(t)})$ is a pruned recurrent neural network satisfying $\|W - W^{sp}\| \leq \delta$. For any time-step $t \geq 1$, define the sequence-of-activity matrix X with iT-th entry $X_{iT} = x_i^{(T)}$, where $i = 1, \ldots, n$ and $T = 1, \ldots, t$. Similarly, define X^{sp} with it-th entry $X_{it} = x_i^{(t,sp)}$. If $\boldsymbol{x}^{(0)} = \boldsymbol{x}^{(0,sp)}$, we have the error bound*

$$\|X - X^{sp}\|_F \leq \delta \sqrt{\sum_{T=1}^{t} \left(\sum_{j=0}^{T-1} k^{j+1} \left\|\boldsymbol{x}^{(T-1-j,sp)}\right\| \|W\|^j \right)^2} .$$

PROOF. By the definition of the Frobenius norm, we have

$$\begin{aligned}
\|X - X^{sp}\|_F^2 &= \sum_{T=1}^{t} \sum_{i=1}^{n} \left| x_i^{(T)} - x_i^{(T,sp)} \right|^2 \\
&= \sum_{T=1}^{t} \left\| \boldsymbol{x}^{(T)} - \boldsymbol{x}^{(T,sp)} \right\|^2 && \text{by definition of } \|\cdot\| = \|\cdot\|_2 \\
&\leq \delta^2 \sum_{T=1}^{t} \left(\sum_{j=0}^{T-1} k^{j+1} \left\| \boldsymbol{x}^{(T-1-j,sp)} \right\| \|W\|^j \right)^2 && \text{by Corollary 4.9,}
\end{aligned}$$

so that taking the square root of the inequality achieves the desired result. □

4.8. Discussion

Over the last decade, there has been a large body of work developing around concentration of measure inequalities. Though their applications are wide-ranging, we discuss others' results in matrix sparsification. As noted throughout this chapter, the Bernstein pruning algorithm was largely inspired by the work of Drineas and Zouzias (2011). They applied the square matrix Bernstein inequality to create a pruning algorithm that sampled entries with replacement, and created an updated version of the algorithm in order to successfully prune the matrix in one-pass. We circumvent this issue by sampling without replacement, which naturally allows our algorithm to only need to pass through each entry of the matrix once (see **[29]** for a review on sampling with or without replacement using Matrix concentration inequalities). Both our algorithm and theirs guarantee an expected number of nonzero entries $O(n \log n \frac{\|A\|_F^2}{\epsilon^2})$ (where $n = \max\{M, N\}$ in our rectangular setting). The matrix Bernstein inequality is definitely not the only useful concentration inequality in the context of pruning. Notably, there has been matrix sparsification work done with matrix inequalities of Bernstein, Khintchine, and Chernoff **[18, 60, 62]**. It will be interesting to see how these can be creatively ported into the neural network literature, especially as scholars recognize that they can be generalized to rectangular matrices.

As in **[60]**, we saw that the matrix Chernoff inequality can be applied to a symmetric, diagonally-dominant matrix $A \in \mathbb{R}^{N \times N}$ to arrive at the inequality

$$|\boldsymbol{x}^T (A - A^{sp}) \boldsymbol{x}| \leq \epsilon |\boldsymbol{x}^T A \boldsymbol{x}| \qquad \text{for all } \boldsymbol{x} \in \mathbb{R}^N,$$

and while we did state that this does imply $\|A - A^{sp}\| \leq \epsilon \|A\|$, it is in fact a stronger statement. Indeed, by the Courant-Fischer theorem, we can see that

$$|\lambda_i - \tilde{\lambda}_i| \leq \epsilon |\lambda_i| \qquad \text{for all } i,$$

where λ_i and $\tilde{\lambda}_i$ are the ith eigenvalues of A and A^{sp} respectively. In other words, the spectral similarity guaranteed in Chapter 3 guarantees that each eigenvalue is approximately preserved relative to the size of the original corresponding eigenvalue. On the other hand, our Bernstein pruning algorithm can only guarantee a relative bound of $\|A - A^{sp}\| \leq \epsilon \|A\|$. We can still receive a bound on the distance between ith eigenvalues thanks to Weyl's inequality **[95]**, but it only reveals:

$$|\lambda_i - \tilde{\lambda}_i| \leq \|A - A^{sp}\| \leq \epsilon \|A\| = \epsilon \lambda_{\max},$$

which is a comparably loose bound for smaller eigenvalues. As such, in the context of symmetric, diagonally dominant matrices, noise-prune still provides stronger theoretical results.

Discrete-time recurrent networks with unitary weight matrices, now known as URNNs, have had a recent surge in popularity due to their ability to circumvent the exploding (or vanishing) gradient problem while outperforming LSTMs and other state-of-the-art networks in hard tasks involving long-term dependencies **[2, 72]**. Our results also apply to networks of this architecture, because discrete-time RNNs can be represented as (infinitely) deep feed-forward networks with identical weights and biases each layer. Indeed, it would be interesting to explore how pruned URNN architectures compare to general RNNs in task-dependent contexts, but that is outside the scope of this present study.

Though the idea to work with contractions/Lipschitz functions in the context of neural networks was entirely an idea developed independently, some recent work has been done discussing "contractive" RNNs **[20]**. They require that both the activation function and the connectivity matrix act as contraction mappings, whereas the results in this work only require that the activation function is a contraction mapping (though our results are strengthened in the event that the connectivity matrices act as contractions/unitary matrices, as observed in Corollary 4.8). However, there does not seem to be any work in the literature that considers pruning these contractive RNNs. The work in this chapter shows that this may be a fruitful future direction for research, as well.

Lastly, it would be interesting to explore a pruning algorithm that mixes the sampling probabilities of the results in Chapters 3 and 4. Specifically, the sampling probabilities in noise-prune take into account higher-order information than weights alone thanks to the comb-cov term. On the other hand, the Bernstein algorithm presented here depends on the square of weights rather than simply the magnitude of weights, so it penalizes small weights more heavily (and favors preserving "larger" weights with higher probability). Perhaps an updated noise-prune that emphasizes preserving larger entries would be worth exploring. It is unlikely that this pruning rule would be predicted by any theoretical argument involving current concentration of measure inequalities, but it may be powerful empirically if the results from Chapter 3 are any indication.

4.9. Appendix

4.9.1. Contractions. Some of the usual activation functions used in neuroscience and machine learning are nonlinearities such as ReLu and sigmoids (i.e., the logistic function, arctangent, and hyperbolic tangent). We will be referring to this collection of functions often, so we define $C = \{\text{ReLu}, \sigma, \tan^{-1}, \tanh\}$.

Each of these nonlinearities are originally functions from $\mathbb{R}$ to $\mathbb{R}$, but when used as activation functions in neural networks, they are applied in an element-wise fashion at each neuron. That is, we can consider them as functions from $\mathbb{R}^N$ to $\mathbb{R}^N$ by simply defining $\boldsymbol{g}(\boldsymbol{x}) = (g(x_1), g(x_2), \ldots, g(x_N))$, where $g \in C$. We review the notion of a contraction mapping below.

Definition 4.11. *A contraction $\boldsymbol{g}$ from $\mathbb{R}^N$ to $\mathbb{R}^N$ has the property that there exists some constant k with $0 \leq k \leq 1$ such that*

$$\|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{y})\|_p \leq k \|\boldsymbol{x} - \boldsymbol{y}\|_p ,$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$. The constant k is said to be the Lipschitz constant of $\boldsymbol{g}(\cdot)$.

It is worth noting that some authors reserve the term *contraction* for $0 \leq k < 1$, and consider allowance of $k = 1$ to instead define a *non-expansion*, but $k = 1$ does not affect the results in this chapter. Indeed, it is the case that $k = 1$ for e.g. ReLu and the Logistic function. Thus, to be inclusive of these functions in our definition of contractions, we are best served including $k = 1$ in our definition.

Intuitively, contractions keep the error between two quantities smaller than the error between the two linear quantities themselves. We now present a simple result that allows us to see that the entry-wise nonlinearities typically used in neural networks (which are one-dimensional contractions) stay contractions when we vectorize them.

Proposition 4.12. *If g is a contraction from $\mathbb{R}$ to $\mathbb{R}$ with Lipschitz constant k under $\|\cdot\|_p = |\cdot|$, then $\boldsymbol{g} : \mathbb{R}^N \to \mathbb{R}^N$ is a contraction from $\mathbb{R}^N$ to $\mathbb{R}^N$ with Lipschitz constant k under $\|\cdot\|_p$, where $\boldsymbol{g}(\boldsymbol{x}) = (g(x_1), g(x_2), \ldots, g(x_N))$.*

Proof.

$$\begin{aligned}\|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{y})\|_p^p &= |g(x_1) - g(y_1)|^p + |g(x_2) - g(y_2)|^p + \cdots + |g(x_N) - g(y_N)|^p \\ &\leq k^p|x_1 - y_1|^p + k^p|x_2 - y_2|^p + \cdots + k^p|x_N - y_N|^p \\ &= k\,\|\boldsymbol{x} - \boldsymbol{y}\|_p^p,\end{aligned}$$

where monotonicity of $(\cdot)^p$ and nonnegativity of k is used in the inequality. Raising both sides of this inequality by the power $1/p$ yields the desired result. □

We solely apply this result for the spectral norm ($p = 2$), but state it for general p-norms for any reader that finds this useful.

Below we define some contraction nonlinearities that the results in the chapter apply to. The ReLu function is given by

$$\text{ReLu}(x) = \max(0, x).$$

A sigmoid function is a bounded, differentiable, function from $\mathbb{R}$ to $\mathbb{R}$ that has a non-negative derivative at each point, and has exactly one inflection point **[33]**. A common example of a sigmoid function is the logistic function

$$\sigma(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}.$$

The tanh function also satisfies the definition of a sigmoid function, and it is given by

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x - e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1}.$$

$\tan^{-1}$ is also a sigmoid function.

4.9.2. Sparsity Guarantees in terms of $\|W_F^2\|$. When we apply Theorem 4.1 to a matrix W with an error tolerance of δ, we are guaranteed an expected number of nonzero entries equal to $O(n \log n \, \|W\|_F^2 \, \delta^{-2})$. Note that $\|W\|_F^2 = r\sigma_{avg}^2$, where r is the rank of W and $\sigma_{avg} = \frac{1}{r}\sum_{i=1}^{r} \sigma_i^2$ (where σ_i^2 is the i-th singular value of W). Then, the expected number of nonzero entries is on the order of

$$O(n \log n \, \|W\|_F^2 \, \delta^{-2}) \propto rn \log n \frac{\sigma_{avg}^2}{\delta^2}.$$

Clearly, if $r = O(n)$, this expression grows as $n^2 \log(n)$, which is larger than the number of non-zero entries in the original matrix. As such, the natural matrix candidates for application of Theorem 4.1 seem to be those with $r = O(\log n)$; that is, when W is of logarithmic rank. Such low-rank matrices have cropped up in statistics and machine learning throughout recent years [**93**].

Bibliography

[1] R. Ahlswede and A. Winter, *Strong converse for identification via quantum channels*, IEEE Transactions on Information Theory, 48 (2002), pp. 569–579.

[2] M. Arjovsky, A. Shah, and Y. Bengio, *Unitary evolution recurrent neural networks*, in Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16, JMLR.org, 2016, p. 1120–1128.

[3] M. Babaeizadeh, P. Smaragdis, and R. H. Campbell, *NoiseOut: A simple way to prune neural networks*, arXiv preprint arXiv:1611.06211, (2016).

[4] M. Baboulin, D. Becker, G. Bosilca, A. Danalis, and J. Dongarra, *An efficient distributed randomized algorithm for solving large dense symmetric indefinite linear systems*, Parallel Computing, 40 (2014), pp. 213–223.

[5] P. Baldi, *Deep Learning in Science*, Cambridge University Press, 2021.

[6] J. Batson, D. A. Spielman, N. Srivastava, and S.-H. Teng, *Spectral sparsification of graphs: theory and algorithms*, Communications of the ACM, 56 (2013), pp. 87–94.

[7] C. Baykal, L. Liebenwein, I. Gilitschenski, D. Feldman, and D. Rus, *Data-dependent coresets for compressing neural networks with applications to generalization bounds*, in 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net, 2019.

[8] C. Baykal, L. Liebenwein, I. Gilitschenski, D. Feldman, and D. Rus, *Sipping neural networks: Sensitivity-informed provable pruning of neural networks*, 2021.

[9] G. Bellec, D. Kappel, W. Maass, and R. A. Legenstein, *Deep rewiring: Training very sparse deep networks*, in 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net, 2018.

[10] D. Blalock, J. J. G. Ortiz, J. Frankle, and J. Guttag, *What is the state of neural network pruning?*, arXiv preprint arXiv:2003.03033, (2020).

[11] P. Borwein and M. Mossinghoff, *Rudin-Shapiro-like polynomials in L_4*, Math. Comp., 69 (2000), pp. 1157–1166.

[12] P. B. Borwein and R. Ferguson, *A complete description of Golay pairs for lengths up to 100*, Math. Comput., 73 (2004), pp. 967–985.

[13] A. G. Burr, *Codes for spread spectrum multiple access systems*, in IEEE International Symposium on Spread Spectrum Techniques and Applications, 1990, pp. 109–115.

[14] J. A. Davis and J. Jedwab, *Peak-to-mean power control in OFDM, Golay complementary sequences, and Reed-Muller codes*, IEEE Trans. Inform. Theory, 45 (1999), pp. 2397–2417.

[15] A. Destexhe, M. Rudolph, and D. Paré, *The high-conductance state of neocortical neurons in vivo*, Nature Reviews Neuroscience, 4 (2003), pp. 739–751.

[16] A. Destexhe and M. Rudolph-Lilith, *Neuronal Noise*, vol. 8, Springer Science & Business Media, 2012.

[17] X. Dong, S. Chen, and S. Pan, *Learning to prune deep neural networks via layer-wise optimal brain surgeon*, in Advances in Neural Information Processing Systems, 2017, pp. 4857–4867.

[18] P. Drineas and A. Zouzias, *A note on element-wise matrix sparsification via a matrix-valued bernstein inequality*, Information Processing Letters, 111 (2011), pp. 385–389.

[19] D. Ž. Đoković, *Equivalence classes and representatives of Golay sequences*, Discrete Math., 189 (1998), pp. 79–93.

[20] M. Emami, M. Sahraee-Ardakan, S. Rangan, and A. K. Fletcher, *Input-output equivalence of unitary and contractive rnns*, CoRR, abs/1910.13672 (2019).

[21] A. A. Faisal, L. P. Selen, and D. M. Wolpert, *Noise in the nervous system*, Nature Reviews Neuroscience, 9 (2008), pp. 292–303.

[22] J. Frankle and M. Carbin, *The lottery ticket hypothesis: Finding sparse, trainable neural networks*, in 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net, 2019.

[23] C. W. Gardiner et al., *Handbook of Stochastic Methods*, vol. 3, Springer Berlin, 1985.

[24] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, *Convolutional sequence to sequence learning*, in International conference on machine learning, PMLR, 2017, pp. 1243–1252.

[25] A. Ghosh, S. Boyd, and A. Saberi, *Minimizing effective resistance of a graph*, SIAM Review, 50 (2008), pp. 37–66.

[26] M. J. E. Golay, *Static multislit spectrometry and its application to the panoramic display of infrared spectra*, J. Opt. Soc. Am., 41 (1951), pp. 468–472.

[27] M. J. E. Golay, *Complementary series*, IRE Trans., IT-7 (1961), pp. 82–87.

[28] S. W. Golomb, *Shift register sequences – a retrospective account*, in Sequences and Their Applications – SETA 2006, G. Gong, T. Helleseth, H.-Y. Song, and K. Yang, eds., Berlin, Heidelberg, 2006, Springer Berlin Heidelberg, pp. 1–4.

[29] D. Gross and V. Nesme, *Note on sampling without replacing from a finite collection of matrices*, ArXiv, abs/1001.2738 (2010).

[30] J. Gu, Z. Lu, H. Li, and V. O. Li, *Incorporating copying mechanism in sequence-to-sequence learning*, arXiv preprint arXiv:1603.06393, (2016).

[31] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, *Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit*, Nature, 405 (2000), pp. 947–951.

[32] R. H. Hahnloser and H. S. Seung, *Permitted and forbidden sets in symmetric threshold-linear networks*, in Advances in Neural Information Processing Systems, 2001, pp. 217–223.

[33] J. Han and C. Moraga, *The influence of the sigmoid function parameters on the speed of backpropagation learning*, From Natural to Artificial Neural Computation. IWANN 1995. Lecture Notes in Computer Science, 930.

[34] S. Han, J. Pool, J. Tran, and W. Dally, *Learning both weights and connections for efficient neural network*, in Advances in Neural Information Processing Systems, 2015, pp. 1135–1143.

[35] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, *Neuroscience-Inspired artificial intelligence*, Neuron, 95 (2017), pp. 245–258.

[36] B. Hassibi and D. G. Stork, *Second order derivatives for network pruning: Optimal Brain Surgeon*, in Advances in Neural Information Processing Systems, 1993, pp. 164–171.

[37] A. J. Holtmaat, J. T. Trachtenberg, L. Wilbrecht, G. M. Shepherd, X. Zhang, G. W. Knott, and K. Svoboda, *Transient and persistent dendritic spines in the neocortex in vivo*, Neuron, 45 (2005), pp. 279–291.

[38] P. R. Huttenlocher, *Synaptic density in human frontal cortex - developmental changes and effects of aging*, Brain Res., 163 (1979), pp. 195–205.

[39] M. Jang, S. Seo, and P. Kang, *Recurrent neural network-based semantic variational autoencoder for sequence-to-sequence learning*, Information Sciences, 490 (2019), pp. 59–73.

[40] L. Jing, Y. Shen, T. Dubcek, J. Peurifoy, S. Skirlo, Y. LeCun, M. Tegmark, and M. Soljačić, *Tunable efficient unitary neural networks (eunn) and their application to rnns*, in International Conference on Machine Learning, PMLR, 2017, pp. 1733–1741.

[41] K. H. A. Kärkkäinen, *Mean-square cross-correlation as a performance measure for department of spreading code families*, in IEEE Second International Symposium on Spread Spectrum Techniques and Applications, 1992, pp. 147–150.

[42] H. Kasai, M. Fukuda, S. Watanabe, A. Hayashi-Takagi, and J. Noguchi, *Structural dynamics of dendritic spines in memory and cognition*, Trends in Neurosciences, 33 (2010), pp. 121–129.

[43] D. J. Katz, *Sequences with low correlation*, in Arithmetic of Finite Fields, L. Budaghyan and F. Rodríguez-Henríquez, eds., Cham, 2018, Springer International Publishing, pp. 149–172.

[44] D. J. Katz, S. Lee, and S. A. Trunov, *Crosscorrelation of Rudin-Shapiro-like polynomials*, Appl. Comput. Harmon. Anal., 48 (2020), pp. 513–538.

[45] ——, *Rudin-Shapiro-like sequences with maximum asymptotic merit factor*, IEEE Trans. Inform. Theory, 66 (2020), pp. 7728–7738.

[46] A. Kazakov and I. Nelken, *Acoustic calibration in an echoic environment*, Journal of Neuroscience Methods, 309 (2018), pp. 60–70.

[47] D. J. Klein and M. Randić, *Resistance distance*, Journal of Mathematical Chemistry, 12 (1993), pp. 81–95.

[48] C. Koch, *Biophysics of Computation: Information Processing in Single Neurons*, Oxford University Press, 2004.

[49] Y. LeCun, J. S. Denker, and S. A. Solla, *Optimal brain damage*, in Advances in Neural Information Processing Systems, 1990, pp. 598–605.

[50] N. Lee, T. Ajanthan, and P. H. S. Torr, *Snip: single-shot network pruning based on connection sensitivity*, in 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net, 2019.

[51] D. Levenstein, B. O. Watson, J. Rinzel, and G. Buzsáki, *Sleep regulation of the distribution of cortical firing rates*, Current Opinion in Neurobiology, 44 (2017), pp. 34–42.

[52] M. Lezcano-Casado and D. Martinez-Rubio, *Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group*, in International Conference on Machine Learning, PMLR, 2019, pp. 3794–3803.

[53] L. Liebenwein, C. Baykal, H. Lang, D. Feldman, and D. Rus, *Provable filter pruning for efficient neural networks*, in International Conference on Learning Representations, 2020.

[54] Z. L. Liu and Y. L. Guan, *Meeting the Levenshtein bound with equality by weighted-correlation complementary set*, in 2012 IEEE International Symposium on Information Theory Proceedings, 2012, pp. 1010–1013.

[55] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, *Multi-task sequence to sequence learning*, arXiv preprint arXiv:1511.06114, (2015).

[56] E. Malach, G. Yehudai, S. Shalev-shwartz, and O. Shamir, *Proving the lottery ticket hypothesis: Pruning is all you need*, in Proceedings of the 37th International Conference on Machine Learning, ICML'20, JMLR.org, 2020.

[57] Z. Mariet and S. Sra, *Diversity networks: Neural network compression using determinantal point processes*, arXiv preprint arXiv:1511.05077, (2015).

[58] W. S. McCulloch and W. Pitts, *A logical calculus of the ideas immanent in nervous activity*, The bulletin of mathematical biophysics, 5 (1943), pp. 115–133.

[59] M. D. McDonnell and L. M. Ward, *The benefits of noise in neural systems: bridging theory and experiment*, Nature Reviews Neuroscience, 12 (2011), pp. 415–425.

[60] E. Moore and R. Chaudhuri, *Using noise to probe recurrent neural network structure and prune synapses*, in Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds., vol. 33, Curran Associates, Inc., 2020, pp. 14046–14057.

[61] S. Narang, G. Diamos, S. Sengupta, and E. Elsen, *Exploring sparsity in recurrent neural networks*, in 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017.

[62] N. H. Nguyen, P. Drineas, and T. D. Tran, *Matrix sparsification via the khintchine inequality*, 2010.

[63] M. G. Parker, K. G. Paterson, and C. Tellambura, *Golay Complementary Sequences*, John Wiley Sons, Ltd, 2003.

[64] T. Paus, M. Keshavan, and J. N. Giedd, *Why do many psychiatric disorders emerge during adolescence?*, Nature Reviews Neuroscience, 9 (2008), pp. 947–957.

[65] Z. Petanjek, M. Judaš, G. Šimić, M. R. Rašin, H. B. Uylings, P. Rakic, and I. Kostović, *Extraordinary neoteny of synaptic spines in the human prefrontal cortex*, Proceedings of the National Academy of Sciences, 108 (2011), pp. 13281–13286.

[66] M. Pievani, N. Filippini, M. P. Van Den Heuvel, S. F. Cappa, and G. B. Frisoni, *Brain connectivity in neurodegenerative diseases—from phenotype to proteinopathy*, Nature Reviews Neurology, 10 (2014), p. 620.

[67] M. B. Pursley, *Performance evaluation for phase-coded spread-spectrum multiple-access communication - part I: System analysis*, IEEE Transactions on Communications, 25 (1977), pp. 795–799.

[68] M. B. Pursley and D. V. Sarwate, *Bounds on aperiodic cross-correlation for binary sequences*, Electronics Letters, 12 (1976), pp. 304–305.

[69] V. Ramanujan, M. Wortsman, A. Kembhavi, A. Farhadi, and M. Rastegari, *What's hidden in a randomly weighted neural network?*, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (2019), pp. 11890–11899.

[70] B. Recht, *A simpler approach to matrix completion*, J. Mach. Learn. Res., 12 (2011), p. 3413–3430.

[71] R. Reed, *Pruning algorithms—a survey*, IEEE transactions on Neural Networks, 4 (1993), pp. 740–747.

[72] A. H. Ribeiro, K. Tiels, L. A. Aguirre, and T. B. Schön, *Beyond exploding and vanishing gradients: analysing rnn training using attractors and smoothness*, 2019.

[73] M. Rudelson, *Random vectors in the isotropic position*, Journal of Functional Analysis, 164 (1999), pp. 60–72.

[74] M. Rudelson and R. Vershynin, *Sampling from large matrices: An approach through geometric functional analysis*, Journal of the ACM (JACM), 54 (2007), p. 21.

[75] W. Rudin, *Some theorems on Fourier coefficients*, Proc. Amer. Math. Soc., 10 (1959), pp. 855–859.

[76] D. Sarwate, *Mean-square correlation of shift-register sequences*, Communications, Radar and Signal Processing, IEE Proceedings F, 131 (1984), pp. 101–106.

[77] D. V. Sarwate and M. B. Pursley, *New correlation identities for periodic sequences*, Electron. Lett., 13 (1977), pp. 48–49.

[78] S. Schug, F. Benzing, and A. Steger, *Presynaptic stochasticity improves energy efficiency and helps alleviate the stability-plasticity dilemma*, eLife, 10 (2021), p. e69884.

[79] S. S. Seiden, *Online randomized multiprocessor scheduling*, Algorithmica, 28 (2000), pp. 173–216.

[80] H. S. Shapiro, *Extremal problems for polynomials and power series*, Master's thesis, Massachusetts Institute of Technology, Cambridge, 1951.

[81] W. R. Softky and C. Koch, *The highly irregular firing of cortical cells is inconsistent with temporal integration of random epsps*, Journal of Neuroscience, 13 (1993), pp. 334–350.

[82] S. Song, P. J. Sjöström, M. Reigl, S. Nelson, and D. B. Chklovskii, *Highly nonrandom features of synaptic connectivity in local cortical circuits*, PLoS Biology, 3 (2005).

[83] D. A. Spielman and N. Srivastava, *Graph sparsification by effective resistances*, SIAM Journal on Computing, 40 (2011), pp. 1913–1926.

[84] D. A. Spielman and S.-H. Teng, *Spectral sparsification of graphs*, SIAM Journal on Computing, 40 (2011), pp. 981–1025.

[85] S. Srinivas and R. V. Babu, *Data-free parameter pruning for deep neural networks*, arXiv preprint arXiv:1507.06149, (2015).

[86] I. S. Stein and K. Zito, *Dendritic spine elimination: molecular mechanisms and implications*, The Neuroscientist, 25 (2019), pp. 27–47.

[87] I. Sutskever, O. Vinyals, and Q. V. Le, *Sequence to sequence learning with neural networks*, Advances in neural information processing systems, 27 (2014).

[88] G. Tononi and C. Cirelli, *Sleep and the price of plasticity: from synaptic and cellular homeostasis to memory consolidation and integration*, Neuron, 81 (2014), pp. 12–34.

[89] H. L. Trentelman, A. A. Stoorvogel, and M. Hautus, *Control Theory for Linear Systems*, Springer Science & Business Media, 2012.

[90] J. A. Tropp, *User-friendly tail bounds for sums of random matrices*, Foundations of Computational Mathematics, 12 (2012), pp. 389–434.

[91] G. G. Turrigiano, *The dialectic of Hebb and homeostasis*, Philosophical Transactions of the Royal Society B: Biological Sciences, 372 (2017), p. 20160258.

[92] R. J. Turyn, *Hadamard matrices, Baumert-Hall units, four-symbol sequences, pulse compression, and surface wave encodings*, J. Combinatorial Theory Ser. A, 16 (1974), pp. 313–333.

[93] M. Udell and A. Townsend, *Why are big data matrices approximately low rank?*, SIAM Journal on Mathematics of Data Science, 1 (2019), pp. 144–160.

[94] M. P. van den Heuvel and A. Fornito, *Brain networks in schizophrenia*, Neuropsychology Review, 24 (2014), pp. 32–48.

[95] R. Vershynin, *High-Dimensional Probability: An Introduction with Applications in Data Science*, no. 47 in Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press.

[96] S. Wass, *Distortions and disconnections: disrupted brain connectivity in autism*, Brain and Cognition, 75 (2011), pp. 18–28.

[97] S. Wisdom, T. Powers, J. Hershey, J. Le Roux, and L. Atlas, *Full-capacity unitary recurrent neural networks*, in Advances in Neural Information Processing Systems, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds., vol. 29, Curran Associates, Inc., 2016.