THE UNIVERSITY OF CHICAGO

SYMMETRIZED FOURIER ANALYSIS OF CONVEX RELAXATIONS FOR COMBINATORIAL OPTIMIZATION PROBLEMS

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY
CHRIS JONES

CHICAGO, ILLINOIS
AUGUST 2022

Dedicated to the young researchers who, like all of us, are just getting started.

TABLE OF CONTENTS

LIST OF FIGURES vii

LIST OF NOTATION ix

ACKNOWLEDGMENTS x

ABSTRACT xi

1 INTRODUCTION 1
1.1 Summary of contributions 3
1.2 Bibliographic note 4
1.3 Outline of the thesis 5
1.4 Basics of Fourier Analysis 5

2 GRAPH MATRICES 8
2.1 Overview of graph matrices 8
2.2 Norm Bounds 13
2.2.1 Sparse norm bounds 16
2.3 Some technicalities and advice 18
2.3.1 General tips 19
2.3.2 Automorphisms 21
2.3.3 Improper shapes and linearization 22
2.3.4 Unordered sets for matrix indices 23
2.4 Multiplying and factoring graph matrices 24
2.5 Open Problems 27

3 OVERVIEW OF SUM-OF-SQUARES LOWER BOUNDS 29
3.1 The sum-of-squares algorithm 29
3.1.1 Sum-of-squares proofs 33
3.2 Pseudocalibration 37
3.2.1 Satisfying equality constraints exactly 40
3.3 Proving PSD-ness using graph matrices 41
3.4 Open Problems 44

4 LOWER BOUND FOR THE SHERRINGTON-KIRKPATRICK MODEL 45
4.1 Statement of Lower Bounds 46
4.1.1 Related work 48
4.2 Proof Outline 49
4.2.1 Graph matrices 49
4.2.2 Outline of PSD-ness proof 52
4.2.3 Informal sketch of key charging lemmas 56
4.3 Pseudocalibration 61

4.3.1 Gaussian setting pseudocalibration 62
4.3.2 Boolean setting pseudocalibration 65
4.3.3 Unifying the analysis 67
4.4 Proving PSD-ness 67
4.4.1 Handling non-spiders 68
4.4.2 Killing a single spider 71
4.4.3 Proving PSD-ness: killing all the spiders 78
4.4.4 Finishing the proof 81
4.5 Reduction from SK model to PAP 83
4.6 Open Problems 85

5 INNER PRODUCT POLYNOMIALS 87
5.1 Overview 87
5.1.1 Constructing the polynomials 89
5.1.2 Related work 91
5.2 Polynomial Basis for the Gaussian Setting 92
5.2.1 Wick Calculus 92
5.2.2 Definitions of inner product polynomials 93
5.2.3 Formulas and properties 96
5.3 Polynomial Basis for the Spherical Setting 99
5.3.1 Definitions of inner product polynomials 100
5.3.2 Formulas and properties 102
5.3.3 Inner product 103
5.4 Polynomial Basis for the Boolean Setting 114
5.4.1 Formulas using the partition poset 117
5.5 Inversion Formula for Approximate Orthogonality 120
5.6 Open Problems 122

6 LOWER BOUND FOR SPARSE INDEPENDENT SET 123
6.1 Statement of results 124
6.1.1 Related work 125
6.2 Proof Outline 126
6.2.1 p-biased Fourier analysis and graph matrices 126
6.2.2 Modifying pseudocalibration 127
6.2.3 Graph matrix analysis 128
6.2.4 Informal sketch for bounding τ and τ_P 131
6.2.5 Proof of Theorem 6.7 136
6.3 Pseudocalibration with connected truncation 138
6.3.1 The failure of “Just try pseudocalibration" 138
6.3.2 Salvaging the doomed 139
6.4 PSD-ness 142
6.4.1 Proof of Lemma 6.43: formalizing the PSD Decomposition 144
6.4.2 Overview of handling combinatorial factors 148
6.4.3 Factor out Π and edges incident to $U_\alpha \cap V_\alpha$ 152

6.4.4 Conditioning I: reduction to sparse shapes 157
6.4.5 Shifting to shapes 161
6.4.6 Counting shapes with the tail bound function $c(\alpha)$ 164
6.4.7 Statement of main lemmas 165
6.4.8 Conditioning II: Frobenius norm trick 168
6.4.9 Norm bounds 169
6.4.10 Proof of main lemmas 171
6.5 Open Problems 175

7 CODE UPPER BOUNDS 177
7.1 Preliminaries 178
7.1.1 Definitions 178
7.1.2 Problem background 179
7.2 Definition of the Hierarchy 181
7.3 Alternative Formulations of the Hierarchy 184
7.3.1 Symmetrization of convex programs 184
7.3.2 Higher-order Kravchuk polynomial hierarchy 185
7.3.3 Subspace symmetrized hierarchy 188
7.3.4 The Hierarchy as an SDP 190
7.3.5 The Hierarchy as ϑ' 192
7.3.6 Comparison with the Sum-of-Squares Hierarchy 193
7.4 Main Properties of the Kravchuk Hierarchies 195
7.4.1 Completeness for Linear Codes 196
7.4.2 Hierarchy Collapse for General Codes 199
7.5 Dual Program 200
7.6 Open Problems 205

REFERENCES 207

LIST OF FIGURES

1.1 A 10-vertex graph. The red vertices are a maximum-size set of vertices such that no two are adjacent. 1

1.2 Left: a 10-vertex graph. Right: By rearranging the vertices, we can see that the graph has two hidden communities. There are many more edges inside the communities than between the communities. 2

2.1 The ribbon $A_R = (1,3), B_R = (2,4), E(R) = \{\{1,2\},\{2,3\},\{3,5\},\{4,5\}\}$. The "left side" is in red, the "right side" is in blue (ordered from top to bottom), and the remaining "middle vertex" is 4. This ribbon corresponds to the Fourier character $\chi_{1,2}(X)\chi_{2,3}(X)\chi_{3,4}(X)\chi_{4,5}(X)$ in the $((1,3),(2,5))$ entry of the matrix. 10

2.2 Left and middle: Two ribbons with the same shape. Right: The "shape" as an unlabeled ribbon. 10

2.3 14

2.4 The purple vertex lies in $U_\alpha \cap V_\alpha$. For each choice of a label k for the purple vertex, for $i,j \in [n]$, the $((i,k),(j,k))$ entries of the graph matrix are essentially a copy of the one-edge shape with $U_\alpha \cap V_\alpha$ deleted. The $((i,k),(j,k))$ entry is $\chi_{\{i,j\}}(X)$ if i,j,k are distinct and 0 otherwise. 20

2.5 23

2.6 (First equality) The left side sums over all i,j,k. The first term on the right side has terms where $i \neq j$, and the second term has terms where $i = j$ (purple denotes $U_\alpha \cap V_\alpha$). The second term is an improper shape, since when $i = j$, both edges become $\{i,k\}$. (Second equality) Since A has Boolean entries, linearize $A_{i,k}^2 = 1$. (Third equality) The isolated vertex scales the matrix by $n-1$. 24

2.7 26

4.1 The Fourier polynomial $h_3(d_{u,i_1})h_1(d_{u,i_2})h_2(d_{u,w_1})h_1(d_{u,j_1})h_1(d_{u,j_2})$ represented as a graph. 50

4.2 Picture of basic non-spider shape α. 52

4.3 Picture of basic spider shape α. 53

4.4 Picture of shapes β_1 and β_2. 54

4.5 Approximation $\beta_1 \times \beta_1^\intercal \approx \alpha$. 55

4.6 Allocation of $\frac{1}{\sqrt{n}}$ from each edge to vertices. S consists of the gray square. Observe that every circle receives a factor of at least $\frac{1}{\sqrt{m}}$ and every square not in S receives a factor of at least $\frac{1}{\sqrt{n}}$. 59

4.7 60

4.8 The five shapes that make up L_4. 73

5.1 89

5.2 Left: Unrouted graph with dashed edges denoting the partial matching collection. Right: Result of routing. 95

5.3 104

5.4 108
5.5 108
5.6 110
5.7 $\mathrm{gg}(C) = \{2\}$. 112
5.8 In the left circle, $\{1, 2, 3\}$ is a non-crossing subset of C. The right four circles show the result of the induced rematching of $\{1, 2, 3\}$. 113

6.1 Dashed edges denote edges in $E_{phantom}$. The 6 vertices that are only incident to dashed edges are isolated in β. Note that $S = \emptyset$ is a separator for β. 135
6.2 The shape from Fig. 6.1 after contracting edge pairs that have been charged to isolated vertices. 135
6.3 The two ribbons on the left side can be added together to get a multiple of the indicator function for the edge $\{1, 2\}$, which is the red edge on the right. However, the leftmost ribbon does not appear due to the connected truncation. 153
6.4 Example of a permissible shape. The solid red edges are independent set indicators. The dashed red edges represent two quasi-missing edge indicators between $U_\alpha \cap V_\alpha$ and the two “floating” connected components of $V(\alpha) \setminus (U_\alpha \cap V_\alpha)$. Not shown are missing edge indicators between all pairs of vertices in U_α and all pairs of vertices in V_α. 156

7.1 203

LIST OF NOTATION

General notation

$\mathsf{X}, \mathsf{X}_1, \mathsf{X}_2, \ldots$	Formal variables
$\mathrm{O}(\cdot), \Omega(\cdot), \mathrm{o}(\cdot), \omega(\cdot)$	Standard asymptotic notation, with respect to $n \to \infty$
n	Input size parameter

Matrix operations

$A \bullet B$	Frobenius inner product
$A \odot B$	Hadamard product
$A \succeq 0$	A is positive-semidefinite

Probability

$x \in_{\mathrm{R}} \Omega$	x is drawn from the uniform distribution on set Ω
i.i.d.	Independent and identically distributed
w.h.p.	With high probability, i.e. probability $1 - \mathrm{o}(1)$

Sets

$\binom{A}{k}$	Collection of subsets of A with size k
$\left(\!\!\binom{A}{k}\!\!\right)$	Collection of sub-multisets of A with k elements
$\mathrm{Sym}(A)$	Group of permutations of A
$S \sqcup T$	Disjoint union of S and T (a multiset including duplicates if necessary)

ACKNOWLEDGMENTS

Acknowledgments often read as a laundry list of shoutouts, and this one is no different. That being said, listing a name is woefully insufficient to describe the depth of experience that I've shared with so many different people, the many discussions we've had, the things we've done together, and the small pleasures of everyday life that have carried me into and through my PhD. If your name appears here (but also if it doesn't!), I am deeply grateful for the interactions we've had over the years.

Thanks to my good friends who have entertained me in various ways. I'll just give you an adjective each. Thanks to the kind Yongshan Ding, the seeking Thomson Yeh, the entertaining Matt McPartlon, the puzzling Patrick Xia, the timeless Nathan Mull, the spry Goutham Rajendran, the outgoing Marie Kim, Jesse "board games" Stern, and to my Galactic friends in the puzzle hunt community. Thanks also to my office mates, the vigilant Neng Huang, the worldly Tushant Mittal, the amiable Reza Jokar, the tireless Hy Truong Son, the sensible Jafar Jafarov, and to the enlightening Zihan Tan.

Relevant to the topic of computer science, most of the work in this thesis is based on collaborative scientific efforts, and it could not have been accomplished without the dedication and insight of my collaborators. Our "sum-of-squares group" in Chicago has consisted of Goutham Rajendran, Fernando Granha Jeronimo, Mrinal Ghosh, Jeff Xu, and Madhur Tulsiani, with Aaron Potechin as the chair. Ideas from these collaborations comprise the content of Chapters 2, 3, 4, and 6. Chapter 5 is joint work with Aaron Potechin. Chapter 7 is joint work with Leonardo Nagami Coregliano and Fernando Granha Jeronimo. A special mention goes to my friend and collaborator Goutham Rajendran for the many hours we've worked together.

I'm grateful for the guidance of Andy Drucker at the start of my PhD, who got me on my research feet. I've been a teaching assistant for Gerry Brady ten times (often with Andrew Eckhart and Matt McPartlon), and I appreciate her commitment to both me and to the students, and the freedom she granted me that made TAing fun. Other academic wisdom has come from Madhur Tulsiani and Laci Babai, who I also thank greatly for their service on various committees. Finally, it almost goes without saying that I am grateful for my advisor, Aaron Potechin. His intuition for problem solving and his honest kindness continue to be an inspiration.

The bedrock without which nothing else could have been possible is my family, including my siblings Doug, Alyssa, and Greg, my parents Sharon and Ken, and my wonderful partner Esther.

ABSTRACT

Convex relaxations are a central tool in modern algorithm design, but mathematically analyzing the performance of a convex relaxation has remained difficult. We develop Fourier analytic methods for the study of convex relaxations, with special focus on semidefinite programming and the sum-of-squares hierarchy.

The sum-of-squares hierarchy is a meta-algorithm for polynomial optimization that has led to recent breakthroughs in combinatorial optimization and robust statistics. We are mostly concerned with lower bounds against sum-of-squares: when does it fail to solve a problem? Barak et al [BHK$^+$16] pioneered the use of Fourier analysis to prove lower bounds against sum-of-squares on average-case problems. We significantly extend their methods, namely *pseudocalibration* and *graph matrix analysis*, helping us to understand when and how the sum-of-squares algorithm succeeds or fails.

We prove concrete lower bounds against sum-of-squares for two problems. First, we prove that sum-of-squares cannot distinguish between: (1) $m = n^{1.5-\varepsilon}$ independently sampled Gaussian vectors in $\mathbb{R}^n$, (2) there is a hidden vector h such that the m vectors are additionally constrained to satisfy $\langle v_i, h \rangle^2 = 1$. Via a reduction, this implies a lower bound for certifying ground states of the Sherrington-Kirkpatrick model, which is a central model in statistical physics. On a technical level, the main challenge is to handle "hard constraints" in the problem; we introduce several new techniques for studying sum-of-squares in the presence of polynomial equality constraints.

Second, we prove that sum-of-squares cannot certify the size of the maximum independent set in a sparse random graph. The main challenge here is sparsity; our work is the first to prove sum-of-squares lower bounds in the sparse regime. We additionally show how to overcome the failure of pseudocalibration in this setting by making a *connected truncation* that gets rid of "global tests".

Moving beyond the setting of the sum-of-squares hierarchy, we introduce a new basis of *inner product polynomials*. For a collection of random vectors $d_1, \ldots, d_m \in \mathbb{R}^n$, these are an orthogonal basis for functions that are orthogonally invariant, meaning that the function only depends on the angles (inner products) between the d_i. These polynomials have many beautiful combinatorial properties related to the topology of an underlying graph.

Finally, we investigate convex programming relaxations for a fundamental problem in the theory of error-correcting codes, the largest possible rate of a code with linear distance. The best upper bound on this quantity was obtained by constructing dual solutions to Delsarte's linear program. We generalize Delsarte's linear program to a hierarchy of linear programs (related to but simpler than the sum-of-squares hierarchy). The hierarchy is similar in structure to Delsarte's linear program, thus offering hope that its value can be theoretically analyzed to prove new upper bounds, although this remains ongoing work.

Thesis Advisor

Aaron Potechin
Assistant Professor, University of Chicago

Committee Members

László Babai
Professor, University of Chicago

Aaron Potechin
Assistant Professor, University of Chicago

Madhur Tulsiani
Associate Professor, Toyota Technological Institute at Chicago
Associate Professor (part time), University of Chicago

Bibliographic note: there are typographic differences between this version of this thesis and the official published version.

CHAPTER 1
INTRODUCTION

We are interested in understanding the complexity of *algorithmic tasks* for given *instances*. For example:

- Given a collection of objects with different sizes and values, and given a fixed-size backpack, what is the highest total value you can fit in the pack? (The knapsack problem)
- Given a graph, what is the largest set of vertices such that no two are adjacent to each other? (The independent set problem, Fig. 1.1)
- Given either a random graph, or a random graph with two communities, can you tell which case you are in? (The stochastic block model, Fig. 1.2)

However, an abundance of questions arises. What algorithm should you use? What is the time/resource complexity of the chosen algorithm? Is it easier to approximately solve your task instead of exactly? What about if your input data is corrupted, distributed, sensitive, or partially hidden? These questions are what keep computer scientists employed.

An important idea that has emerged is the use of *black box*, or *metaprogramming*, algorithms. To use a metaprogramming algorithm, we plug our problem into an algorithmic black box without much thinking, then the black box spits out a candidate output (although usually in practice there is still some thinking needed). Despite the lack of ingenuity required, we have formal and informal evidence that a metaprogramming algorithm is the *best possible choice* of algorithm for certain types of algorithmic tasks! If any algorithm works, then this particular algorithm must work. (It is possible that the task is computationally intractable and all efficient algorithms, including the metaprogramming algorithm, will fail.)

In this thesis we focus on *convex programming algorithms*, with special emphasis on the *sum-of-squares* metaprogramming algorithm. The importance of convex programming as a general technique cannot be overstated. Convex programs are important both in theory

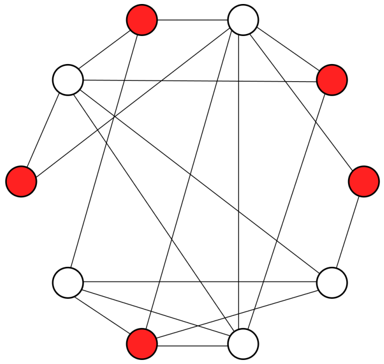


Figure 1.1: A 10-vertex graph. The red vertices are a maximum-size set of vertices such that no two are adjacent.

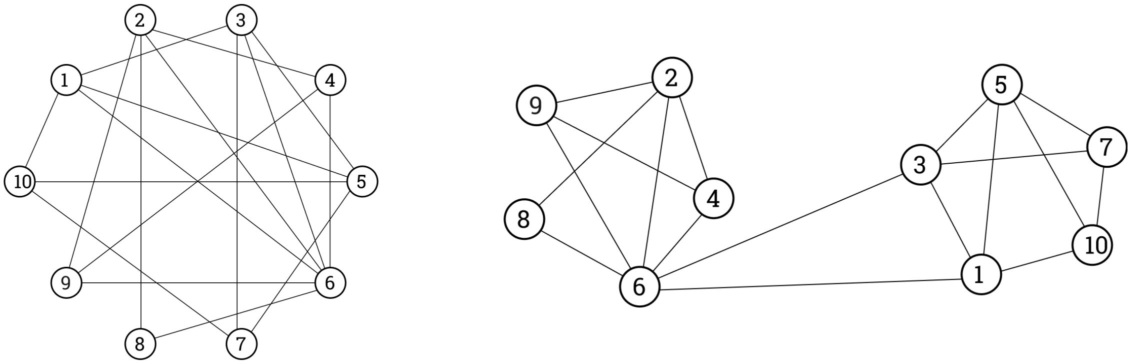


Figure 1.2: Left: a 10-vertex graph. Right: By rearranging the vertices, we can see that the graph has two hidden communities. There are many more edges inside the communities than between the communities.

and practice, on the one hand underlying many if not most of our best approximation algorithms, and on the other hand being the algorithm of choice for large-scale industrial logistics problems such as airline scheduling and package delivery.

The sum-of-squares algorithm is a generic and powerful convex programming meta-algorithm. Sum-of-squares can be run as a black box for polynomial optimization (although, again, some algorithmic design choices may be necessary or helpful). We have evidence that sum-of-squares is optimal among a large class of convex programming algorithms for certain *combinatorial optimization problems* [LRS15], and in fact further evidence that convex programming is itself the optimal algorithm for these problems [Rag08]. In the last decade, the sum-of-squares algorithm has led to numerous breakthroughs in combinatorial optimization and computational statistics. Examples and more background on sum-of-squares are given in Chapter 3.

Despite the fact that we can set up and run the sum-of-squares algorithm as a black box, we cannot determine whether or not the algorithm solves our problem! From an empirical standpoint, the sum-of-squares algorithm is largely a theoretical algorithm because its runtime is too slow on nontrivial inputs. By running the code, we can assess whether the algorithm solves a particular instance, though we cannot extrapolate the behavior to all inputs, because (1) the inputs that can be feasibly computed are relatively small, on the order of kilobytes; (2) there may be nefarious inputs that "thwart" our algorithm that we have failed to consider.

From a theoretical standpoint, even our mathematical understanding of convex relaxations such as sum-of-squares remains quite limited. We don't know how to theoretically "run" a convex relaxation on a large instance and predict the output. We would like a mathematical analysis of exactly how well the algorithm performs, so that we can be 100% convinced one way or the other of the quality of the output. However, mathematical proofs remain few, and the cases in which we have a theoretical analysis of a convex relaxation are vastly outnumbered by the cases in which we only have predictions. Because of the state-of-the-art applications of sum-of-squares and other convex programming relaxations, a better understanding of these algorithms is undoubtedly important for future algorithmic research.

The primary task of this thesis is to theoretically analyze convex programming relaxations such as sum-of-squares. We prove *lower bounds* against the algorithm in situations where it fails, i.e. we prove that the algorithm's output is significantly worse than optimal. Based on the strength of the sum-of-squares metaprogramming algorithm, such lower bounds provide strong, concrete evidence that *there is no efficient algorithm for the problem*. Another important aspect of our work is that it yields structural insights into sum-of-squares, helping us to understand when and how the algorithm succeeds or fails.

For performing the algorithm analysis, we develop and employ new mathematical technology. A portion of the thesis is devoted to explaining and building out these tools. The tools we need belong to the fields of Fourier analysis, random matrix theory, and combinatorics. The methods are likely to be of wider interest in the theoretical computer science and mathematics communities.

1.1 Summary of contributions

We present the specific contributions of the thesis in more detail.

New Fourier analytic methods. An overarching mathematical tool that we utilize is *Fourier analysis*, especially in the presence of *symmetry*. The general idea of Fourier analysis is to decompose a function in a special way (with respect to harmonic functions in the *Fourier basis*) and use the decomposition to reason about the function. This idea has been incredibly fruitful, and the basic tool of Fourier analysis is one of the most important mathematical developments ever. In our settings, the functions under consideration will have additional symmetry properties, and therefore we will use a basis that has the same symmetry.

First, we develop Fourier analysis of functions whose output is a matrix. A basis for decomposing these functions is the space of *graph matrices*. Graph matrices were introduced by Barak et al [BHK$^+$16], and our work significantly extends the scope of graph matrix analysis. For example, we show a combinatorial approach to analyzing the null space of the matrix, and how to use graph matrices in the *sparse setting*, including new *norm bounds* for graph matrices over sparse inputs.

Second, we define and study the space of *inner product functions*. A function in this space takes as input a collection of vectors, and the output is *orthogonally invariant*: it only depends on the angles, or inner products, between the vectors. We define an explicit basis for this space. We derive many beautiful formulas that relate properties of the basis to combinatorial and topological properties of the graphs that define the basis.

Sum-of-squares lower bounds. Our Fourier analytic techniques were invented to attack the main challenge of the thesis, proving lower bounds against the *sum-of-squares algorithm*. We prove lower bounds for two specific problems. First, we prove that sum-of-squares cannot certify ground states of the *Sherrington-Kirkpatrick model*, which is a central model in statistical physics that exhibits deep mathematical behavior. Second, we prove that sum-of-squares cannot certify the *maximum independent set* in a sparse random graph, which is a fundamental problem in combinatorial optimization.

To analyze the sum-of-squares algorithm on these problems, we need many new sum-of-squares specific ideas, in addition to the Fourier analytic techniques. For the Sherrington-Kirkpatrick model, we show new techniques for handling polynomial equality constraints in the input, such as handling the null space of the moment matrix and rounding the pseudoexpectation operator. For the maximum independent set problem, we extend the *pseudocalibration* technique to decouple the local and global properties of the input graph, and we overcome a host of new challenges related to the sparse setting.

Prior to the last decade, very few sum-of-squares lower bounds were known; it is only in the past ten years that we have started to gain a foothold into the analysis of sum-of-squares. For example, prior to our work, sum-of-squares lower bounds were obtained for problems in the "dense" input regime, while the sparse regime remained out of reach. Our work on independent set is the first progress in this direction.

Applications of convex relaxations in coding theory. In the last part of the thesis, we investigate convex programming techniques for a fundamental problem in the theory of *error-correcting codes*. We propose a new convex relaxation for the size of the largest error-correcting code with given distance. This is a problem that has not seen improvement since the 1970s, and our technique suggests a promising new direction of attack. Our relaxation is similar to but simpler than the sum-of-squares relaxation for this problem, and as we explain, the simplicity is likely to help with analyzing the program. We prove that the relaxation is *complete*, meaning that a sufficiently high "level" of the program exactly solves the problem (and there is an implicit hope that analyzing "low levels" of the algorithm can lead to improved bounds).

A conceptual difference between this research and the other parts of the thesis is that the input is a specific, deterministic instance whereas the input is random in all other chapters. Analyzing convex programs on specific instances can be very difficult, and analyzing the value of our program remains ongoing work, some of which we describe.

1.2 Bibliographic note

The ideas in this thesis were developed through collaboration with several other researchers. These collaborations resulted in four peer-reviewed research papers [GJJ$^+$20, JP22, JPR$^+$21, CJJ22] which together contain the core ideas of the thesis.

In this thesis, the underlying technical tools have been reorganized in order to separate the mathematical technique of "symmetrized Fourier analysis" from its application to the sum-of-squares hierarchy. Furthermore, exposition of the core ideas from the papers has been modified either to improve the presentation or to highlight the author's viewpoint, some additional proofs have been added, and specific contributions of the author have been emphasized. Some important parts of the joint papers are included, even if the author was not the main author of those parts. In the introductory section of each chapter there is a bibliographic attribution for that chapter which explains the differences between the chapter and the relevant published works. Each chapter also ends with related open problems.

The thesis of Rajendran [Raj22] also contains the main material of [GJJ+20].

1.3 Outline of the thesis

Chapter 2 introduces the theory of matrix-valued Fourier analysis, known as *graph matrices*. This chapter is expository and most proofs are elided.

Chapter 3 provides an overview of the sum-of-squares algorithm and describes at a high level our technique for proving lower bounds. This chapter is also expository.

Chapter 4 and Chapter 6 prove the lower bounds against the sum-of-squares algorithm. Chapter 4 deals with the Sherrington-Kirkpatrick model and Chapter 6 deals with independent set on sparse random graphs. These chapters use the graph matrix technology from Chapter 2 and the proof strategy in Chapter 3.

Chapter 5 introduces inner product polynomials, a basis for a particular space of symmetric functions. It was developed (though ultimately not used) as a tool for studying the Sherrington-Kirkpatrick model in Chapter 4. This chapter can be read independently of the other chapters.

Chapter 7 gives the applications of convex relaxations in coding theory. This chapter can be read independently of the other chapters.

The remainder of this introduction will provide a quick overview of Fourier analysis in the presence of symmetry.

1.4 Basics of Fourier Analysis

Fourier analysis expresses a function f with respect to a *Fourier basis* of "harmonic functions", known as *Fourier characters*. There are many types of Fourier analysis, depending on the domain of f and the exact generalization of "harmonic" that one is working with.[1] The situation that underlies all the work in this thesis is that we will have some distribution $\mathcal{D}$ on $\mathbb{R}^N$ and we will think of "functions of a random variable" $f(X)$ where $X \sim \mathcal{D}$. In this case the Fourier basis for f consists of the *orthogonal polynomials* for the distribution $\mathcal{D}$.

For example, the most common distribution encountered in computer science is the uniform distribution on the Boolean hypercube $\{-1,+1\}^n$ (this is essentially equivalent to the uniform distribution on $\{0,1\}^n$).

Definition 1.1. *For a finite set Ω, $x \in_R \Omega$ denotes a sample from the uniform distribution on Ω.*

For input $x \in_R \{-1,+1\}^n$, the Fourier basis for $f : \{-1,+1\}^n \to \mathbb{R}$ consists of the set of multilinear monomials.

[1] In the typical set-up, $f : G \to \mathbb{C}$ where G is a group, and the basis of harmonic functions is group characters.

Definition 1.2. *Let $v \in \mathbb{R}^n$. For $I \subseteq [n]$ or $\alpha \in \mathbb{N}^n$, let $v^I := \prod_{i \in I} v_i$ or $v^\alpha := \prod_{i=1}^n v_i^{\alpha_i}$. The notation also extends to the case when $v = (v_1, \ldots, v_n)$ is a vector of variables.*

Definition 1.3 (Boolean Fourier character)**.** *For $S \subseteq [n]$ and $x \in \{-1, +1\}^n$, $\chi_S(x) = x^S$.*

The functions χ_S are orthonormal under the uniform distribution on the hypercube. We use the notation $\mathbb{1}_{S=T}$ as the $0/1$ indicator of whether $S = T$.

Definition 1.4. *Let $f, g : \{-1, +1\}^n \to \mathbb{R}$. We denote by $\langle f, g \rangle = \mathop{\mathbb{E}}_{x \in_R \{-1,+1\}^n} [f(x)g(x)]$ the inner product of f and g.*

Fact 1.5. *The functions χ_S are orthonormal: $\langle \chi_S, \chi_T \rangle = \mathbb{1}_{S=T}$. Furthermore, they form a basis for all functions $f : \{-1, +1\}^n \to \mathbb{R}$.*

The Fourier transform gives the coefficients of f when expressed in the Fourier basis.

Definition 1.6 (Fourier transform)**.** *The Fourier transform $\widehat{f}$ of $f : \{-1, +1\}^n \to \mathbb{R}$ is defined on $S \subseteq [n]$ by:*

$$\widehat{f}(S) = \langle f, \chi_S \rangle = \mathop{\mathbb{E}}_{x \in_R \{-1,+1\}^n} [f(x) \cdot \chi_S(x)].$$

Fact 1.7 (Fourier inversion)**.** *Let $f : \{-1, +1\}^n \to \mathbb{R}$. Then $f(x) = \sum_{S \subseteq [n]} \widehat{f}(\alpha) \chi_S(x)$.*

Fact 1.8 (Plancherel identity)**.** *Let $f, g \colon \{-1, +1\}^n \to \mathbb{R}$. Then $\langle f, g \rangle = \sum_{S \subseteq [n]} \widehat{f}(S) \cdot \widehat{g}(S)$.*

In the settings that we consider, the function f will additionally satisfy some *symmetry* under a group action. It may be more natural to express f with respect to a basis of functions which also have the symmetry. To give an example, suppose that the output of a function $f : \{-1, +1\}^n \to \mathbb{R}$ only depends on the Hamming weight of the input.

Definition 1.9 (Hamming weight)**.** *For $x \in \{-1, +1\}^n$, the Hamming weight $|x|$ is the number of -1's in x.*

Equivalently, f satisfies the symmetry $f(x) = f(\pi(x))$ for all permutations $\pi \in S_n$, where the group action permutes the bits of x. When f is written in the Fourier basis, the Fourier coefficient $\widehat{f}(S)$ also only depends on $|S|$.

Lemma 1.10. *If $f(x) = f(\pi(x))$ for all permutations $\pi \in S_n$, then $\widehat{f}(S) = \widehat{f}(\pi(S))$ for all S.*

Proof.

$$\begin{aligned}\widehat{f}(S) &= \mathop{\mathbb{E}}_{x\in_{\mathrm{R}}\{-1,+1\}^n}[f(x)\chi_S(x)] \\ &= \mathop{\mathbb{E}}_{x\in_{\mathrm{R}}\{-1,+1\}^n}[f(\pi(x))\chi_S(x)] \\ &= \mathop{\mathbb{E}}_{x\in_{\mathrm{R}}\{-1,+1\}^n}[f(x)\chi_S(\pi^{-1}(x))] \\ &= \mathop{\mathbb{E}}_{x\in_{\mathrm{R}}\{-1,+1\}^n}[f(x)\chi_{\pi(S)}(x)] \\ &= \widehat{f}(\pi(S))\,.\end{aligned}$$

□

Therefore, a basis for f consists of the following functions ψ_i and the following decomposition:

$$\psi_i(x) = \sum_{\substack{S\subseteq[n],\\ |S|=i}} \chi_S(x) \qquad (i = 0, 1, \ldots, n)$$

$$f(x) = \sum_{i=0}^{n} \widehat{f}(i)\psi_i(x)$$

where $\widehat{f}(i)$ is the Fourier coefficient for sets of size i.

The function f can also be expressed purely in terms of the Hamming weight $|x|$, i.e. as a function on the set $\{0, 1, 2, \ldots, n\}$. The functions $\psi_i(x) = K_i(|x|)$ are equal to the *Kravchuk polynomials* K_i evaluated on the Hamming weight of x. The Kravchuk polynomials are the orthogonal polynomials for the binomial distribution. It makes sense that they would appear here, since the distribution of $|x|$ for $x \in_{\mathrm{R}} \{-1, +1\}^n$ is a binomial random variable $\mathrm{Bin}(n, 1/2)$.

Usually, a symmetrized basis can be easily obtained by just grouping together the non-symmetrized Fourier characters, like we did here to get the Kravchuk polynomials from the Boolean Fourier characters. However, that isn't the case in Chapter 5.

CHAPTER 2
GRAPH MATRICES

In this chapter we study Fourier analysis of matrix-valued functions. When the matrix-valued function is *permutation-symmetric*, a Fourier basis is the set of *graph matrices*. Just as in Fourier analysis of scalar-valued functions, we can prove many things about our function by expressing it with respect to the graph matrix basis. Here we will survey without proofs some of the useful properties of the graph matrix basis.

The most useful property for our purposes are *norm bounds* for graph matrices that hold with high probability when the input is chosen randomly. Understanding the spectral norms of random matrices is crucial for controlling error terms in both algorithms and lower bounds.

The graph matrix toolbox described in this chapter is the key mathematical technology underlying the proofs in Chapter 4 and Chapter 6. In those chapters, we apply graph matrices to study the sum-of-squares algorithm.

An overview of graph matrix analysis is given in Section 2.1. The norm bounds are described in Section 2.2. Section 2.3 discusses some technicalities in the definition of graph matrices and offers some helpful tips for using them. Section 2.4 explains the calculus of multiplying and factoring graph matrices.

Unfortunately a one-size-fits-all definition of graph matrices is fairly unwieldy, for some of the reasons listed in Section 2.3. Other chapters that use graph matrices will give the particular definition of graph matrices used in the context of the chapter.

Bibliography. Graph matrices are not an original contribution of this thesis, having been invented in the context of the Planted Clique problem. This includes works by Deshpande and Montanari [DM15] (introduction of ribbons), Meka-Potechin-Wigderson [MPW15] (similar-looking matrices, with norm bounds proved via trace method), Barak et al [BHK$^+$16] (factoring and multiplying graph matrices), and Potechin and Rajendran [PR20] (generalizing the technology beyond Planted Clique). Further background and norm bounds in the "dense case" are given by Ahn-Medarametla-Potechin [AMP20]. Works of the author utilizing and expanding the theory are [GJJ$^+$20, JPR$^+$21]; the latter work introduced the "sparse case" described in Section 2.2.1. The exposition here is for the most part original.

Graph matrices have also been called "graphical matrices" [BHK$^+$16] and "random association schemes" [DM15]. Other reasonable names may be "symmetrized matrix Fourier characters" or "functional association schemes". The graph matrix technology is still being developed, and it is likely that future work will improve their usage beyond what is written here.

2.1 Overview of graph matrices

Given a matrix-valued function $f(X) \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ of a random variable $X \sim \mathcal{D}$, if f is permutation-symmetric (to be defined below) we can express it in a symmetrized Fourier basis, known as *graph matrices* $M_\alpha(X)$.

$$f(X) = \sum_{\text{shapes } \alpha} \widehat{f}(\alpha) M_\alpha(X)\,.$$

The high-level strategy for using graph matrices is:

- Each graph matrix can be specified by a succinct graph α, the *shape*.
- Properties of the graph matrix, notably the spectral norm, are determined by combinatorial properties of the shape.
- Therefore, to analyze a function expressed in the basis of graph matrices, it suffices to reason combinatorially about the shapes.

The running example that we use in this chapter is when X is the Erdős-Rényi random graph $G_{n,1/2}$, with vertex set $[n]$ and each edge included independently with probability $1/2$. The matrix indices in $\mathcal{I}$ are all ordered subsets of $[n]$.

Since the Erdős-Rényi random graph consists of $\binom{n}{2}$ independent bits, the Fourier basis is given by the Boolean Fourier characters. There is a character χ_E for each $E \subseteq \binom{[n]}{2}$. Therefore, the Fourier characters are in correspondence with undirected graphs on $[n]$.

The natural way to extend this to a basis for matrix-valued functions is to express each entry of the matrix as a scalar function in the Fourier basis. A *ribbon* specifies a particular Fourier character in a particular entry of the matrix. It consists of the matrix entry as well as which Fourier character it is.

Definition 2.1 (Ribbon)**.** *A ribbon is a tuple $R = (A_R, B_R, E(R))$ where $A_R, B_R \in \mathcal{I}$ and $E(R)$ specifies the Fourier character. The corresponding matrix $M_R(X) \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ is:*

$$M_R(X)[I,J] = \begin{cases} \chi_{E(R)}(X) & I = A_R, J = B_R \\ 0 & \textit{otherwise}\,. \end{cases}$$

A ribbon is a combinatorial object. We call A_R and B_R the “left” and “right” sides of R. In the case of $G_{n,1/2}$, $E(R)$ defines a small graph; we let $V(R) := A_R \cup B_R \cup V(E(R))$. We also define the set of “middle vertices” $C_R := V(R) \setminus (A_R \cup B_R)$. An example ribbon is in Fig. 2.1.

The ribbon basis is too fine to aptly summarize the matrix properties of $f(X)$. Fortunately, if $f(X)$ is *permutation-symmetric*, we can group ribbons into shapes, which are more effective. In the context of $G_{n,1/2}$, a shape can be thought of as a ribbon where we have erased the vertex labels, as in Fig. 2.2.

Definition 2.2 (Shape for $G_{n,1/2}$)**.** *A shape is an equivalence class of ribbons under relabeling of $[n]$. Each shape has a representative $\alpha = (U_\alpha, V_\alpha, E(\alpha))$.*

We also define the “middle vertices” $W_\alpha = V(\alpha) \setminus (U_\alpha \cup V_\alpha)$, corresponding to C_R. Greek letters are generally used for shapes whereas Latin letters are used for ribbons.

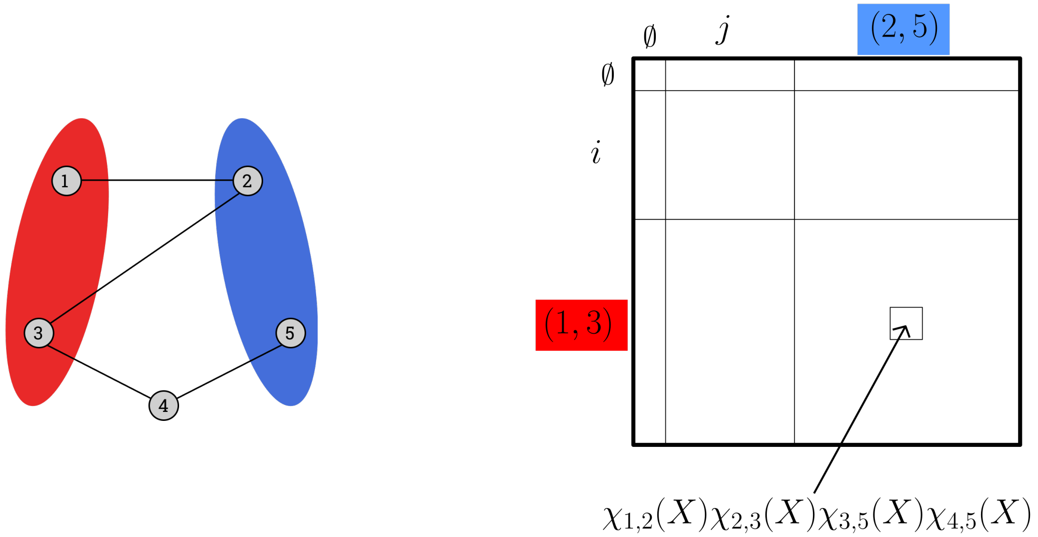


Figure 2.1: The ribbon $A_R = (1,3), B_R = (2,4), E(R) = \{\{1,2\},\{2,3\},\{3,5\},\{4,5\}\}$. The "left side" is in red, the "right side" is in blue (ordered from top to bottom), and the remaining "middle vertex" is 4. This ribbon corresponds to the Fourier character $\chi_{1,2}(X)\chi_{2,3}(X)\chi_{3,4}(X)\chi_{4,5}(X)$ in the $((1,3),(2,5))$ entry of the matrix.

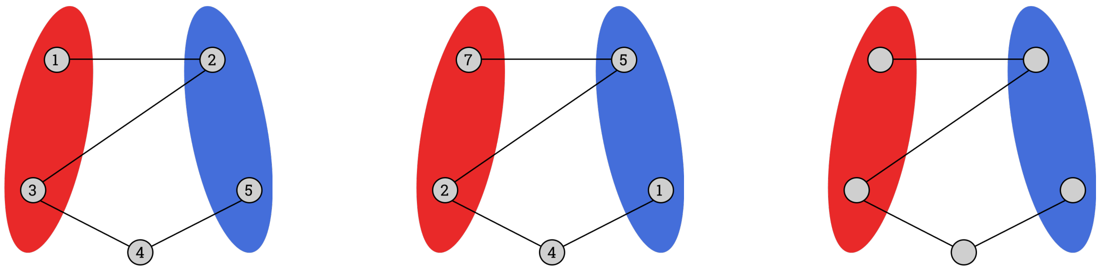


Figure 2.2: Left and middle: Two ribbons with the same shape. Right: The "shape" as an unlabeled ribbon.

Definition 2.3 (Graph matrix). *For a shape α, $M_\alpha(X) = \sum_{\text{ribbon } R \text{ of shape } \alpha} M_R(X)$.*

When $f(X)$ is a permutation-symmetric matrix-valued function, by grouping together ribbons with the same coefficient, it can be expressed as a linear combination of graph matrices $M_\alpha(X)$. It is easier to see the definition of permutation-symmetric through examples first and the definition second. In the context of $G_{n,1/2}$, permutation-symmetric functions $f(X)$ of $X \sim G_{n,1/2}$ do the "same thing" for each entry $f(X)[I,J]$, just "localized" to the vertex sets I, J. The adjacency matrix of X is permutation-symmetric. Other permutation-symmetric functions are $f(X)[i,j]$ = number of triangles containing edge i, j, and $f(X)[(i,j),(k,l)] = 0/1$ if $\{i,j,k,l\}$ are a 4-clique.

In the general setting (not necessarily $G_{n,1/2}$), we have a group acting on both the index set $\mathcal{I}$ and the input X (technically speaking, an action on the underlying probability space). For $G_{n,1/2}$ this is the induced action of the symmetric group S_n on ordered subsets of $[n]$ or graphs on $[n]$.

Definition 2.4 (Permutation-symmetric). *Given a group acting on both $\mathcal{I}$ and X, a matrix-valued function $f(X) \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ is permutation-symmetric if $f(X)[I,J] = f(\pi(X))[\pi(I),\pi(J)]$ for all π in the group.*

Definition 2.5 (Shape). *A shape is an orbit of ribbons under the group actions.*[1] *Each shape has a representative $\alpha = (U_\alpha, V_\alpha, E(\alpha))$.*

Definitionally, we have the following proposition:

Proposition 2.6. *Graph matrices form a basis for permutation-symmetric matrix-valued functions $f(X) \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$.*

Definition 2.7 (Trivial shape). *A shape α is trivial if $U_\alpha = V_\alpha$ and $E(\alpha) = \emptyset$ is the trivial Fourier character.*

$M_\alpha(X)$ for a trivial α is the identity matrix restricted to the block $U_\alpha \times U_\alpha \subseteq \mathcal{I} \times \mathcal{I}$.

Definition 2.8 (Transpose). *Given a ribbon R or shape α, we define its transpose $R^\intercal$ or $\alpha^\intercal$ by swapping A_R and B_R (resp. U_α and V_α). Observe that this transposes the matrix for the ribbon/shape.*

Some remarks on graph matrices:

1. To simplify the notation, the input is usually dropped from the notation. A graph matrix is written M_α instead of $M_\alpha(X)$.

2. In the context of $G_{n,1/2}$, the ribbons of shape α are exactly those that result from embedding the shape graph α into the overall vertex set $[n]$.

   Lemma 2.9. *In the context of $G_{n,1/2}$, a ribbon R has shape α if and only if there is an injective map $\phi : V(\alpha) \to [n]$ such that $\phi(U_\alpha) = A_R$ and $\phi(V_\alpha) = B_R$ as ordered tuples, and $\phi(E(\alpha)) = E(R)$.*

[1] We require that the group action on X preserves Fourier characters, meaning for all E and π there is F such that $\chi_E(\pi(X)) = \chi_F(X)$.

In fact, it is usually simpler to take the definition of a graph matrix to be the sum over embeddings of α. This definition differs from the one we have given by a scalar factor (the size of an automorphism group). See Section 2.3.2 for this and Section 2.3 for further remarks on the definition of a graph matrix.

3. Occasionally, a simpler basis can be used without using the "full strength" of the graph matrix basis. For the example case of $G \sim G_{n,1/2}$, if the index set is $\mathcal{I} = V(G) = [n]$, and the entry of a permutation-symmetric matrix $f(X)[u,v]$ depends only on the graph distance between u and v, then a basis for $f(X)$ is powers A^r of the centered adjacency matrix A. For analyzing the spectrum of $f(X)$, it therefore suffices to analyze the spectrum of A, which is significantly more well-studied than graph matrices in general [Tao12, Bor15].

4. When the input X has several dimensions (for example, X is a random $m \times n$ matrix), a Fourier character may be viewed as a graph in which vertices have different "types" (either "row" or "column"). Usually the function $f(X)$ is permutation-symmetric under independent permutation of each type. This situation is the case in Chapter 4.

5. Even if X is not a random graph, the underlying graph or hypergraph structure of the Fourier character is still relevant to the norm bound. This provides some justification for the general term "graph matrix".

6. To elaborate on the previous point, the settings that we consider will all have $X = (X_1, \dots, X_N) \in \mathbb{R}^N$. A single Fourier character is indexed by $\alpha \in \mathbb{N}^N$ which can be thought of as a hyperedge on $[N]$ in which vertices are allowed to appear more than once. The matrices will always be indexed by $\mathcal{I}$ being sets, multisets, or tuples from $[N]$ (this is the typical setting for the sum-of-squares algorithm, Chapter 3). Therefore, a ribbon is a hypergraph structure on $[N]$ with specialized left and right subsets (or multisets or tuples) of vertices from $[N]$. Passing to shapes, the action on $\mathcal{I}$ and X will be by a permutation group $G \leq S_N$.

7. Graph matrices can be seen as "functional association schemes". We briefly review the definitions. Matrices that are symmetric under a group action (without being functions of X) exactly lie in a corresponding association scheme.

   Definition 2.10 (Schurian association scheme). *Given a transitive permutation group $G \leq \mathrm{Sym}(\mathcal{I})$, the association scheme for G acting on $\mathcal{I}$ is the set of orbits of the induced action of G on $\mathcal{I} \times \mathcal{I}$. In the matrix view, for each orbit in $\mathcal{I} \times \mathcal{I}$, we take a 0/1 matrix in $\mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ indicating the orbit.*

   Fact 2.11. *Given a transitive permutation group $G \leq \mathrm{Sym}(\mathcal{I})$, a matrix $M \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ is G-symmetric if and only if M lies in the linear span of the association scheme for G acting on $\mathcal{I}$.*

Remark 2.12. *Association schemes arising from permutation groups in this way are called "Schurian", whereas the full definition of association scheme is a combinatorial generalization, which just requires combinatorial properties of the underlying 0/1 matrices without the presence of the permutation group.*

In our setting where the matrix is random (a function of X) and G also acts on X, there is one basis matrix per orbit of the action on $\mathcal{I} \times \mathcal{I}$ and X together. Note that this *not* the same as saying that each instantiation of X gives an association scheme.

It should be noted that, because we are doing Fourier analysis, graph matrices use Fourier characters instead of matrix entries that are $0/1$.

2.2 Norm Bounds

Now that we have expressed a matrix-valued function in terms of ribbons and collected the ribbons into graph matrices, we would like to study the individual graph matrices M_α. A common goal in this thesis is to understand the spectrum of a matrix-valued function, with high probability over a random input. It turns out that for different α, the spectra of M_α lie on different scales in terms of n (for a random input), and a first step is to determine the magnitudes.

Definition 2.13 (Spectral norm)**.** *The spectral norm of* $M \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ *is* $\|M\| = \max\limits_{\substack{x\in\mathbb{R}^{\mathcal{I}},\\ x\neq 0}} \dfrac{\|Mx\|_2}{\|x\|_2}$.

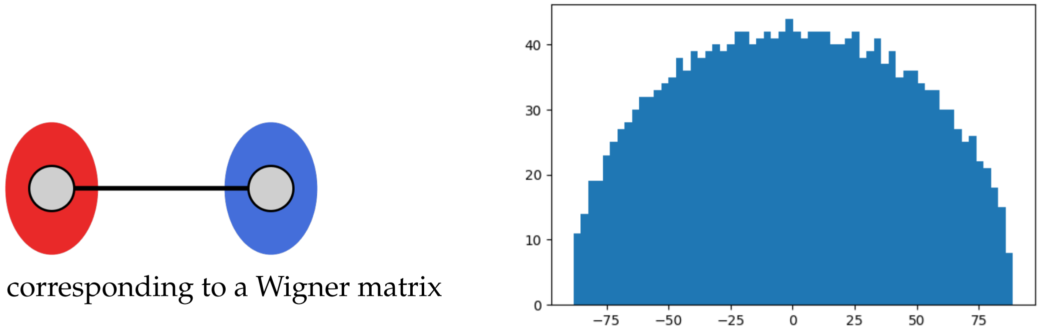
When M is symmetric, $\|M\|$ is the maximum absolute value of the eigenvalues. In general, $\|M\|$ is the maximum singular value.

Example 2.14 (Wigner matrix)**.** *A random symmetric matrix with independent entries is known as a Wigner matrix. Consider a $n \times n$ Wigner matrix with independent Rademacher entries (i.e., uniform from $\{-1, +1\}$). It is well known that the spectrum lies in $[(-2 - o(1))\sqrt{n}, (2 + o(1))\sqrt{n}]$ with high probability, and furthermore that the spectrum converges to the Wigner semicircle law on this interval (in a sense we don't define here). This matrix is described by the one-edge shape in Fig. 2.3 (except for the diagonal entries[2]), and hence the spectral norm of this simple shape is $\Theta(\sqrt{n})$.*

Unlike the Wigner matrix in the previous example, for general α the entries of M_α are *not* independent. They are degree-$|E(\alpha)|$ polynomial functions of the underlying object (however, the underlying object usually does have iid entries).

The magnitude in n of the spectral norm of M_α can be expressed in terms of combinatorial properties of the shape α, namely *vertex separators*. The theorems in this section will be stated for the case where the underlying randomness is $G_{n,1/2}$, and then for the case of $G_{n,p}$. The general case of a set of independent variables with all moments finite and

[2] Since the diagonal entries are at most 1 in absolute value, adding or subtracting the diagonal entries changes the spectral norm by at most 1, which is comparatively small since most eigenvalues are on the scale $\Theta(\sqrt{n})$.



(a) Shape corresponding to a Wigner matrix

(b) Histogram of the eigenvalues of a random 2000×2000 Wigner matrix. The largest eigenvalue magnitude is near $2\sqrt{n} \approx 89.44$.

Figure 2.3

independent of n can be found in [AMP20]. Note that their work generalizes $G_{n,1/2}$ but not $G_{n,p}$.

Definition 2.15 (Minimum vertex separator). *For a shape α, a set $S \subseteq V(\alpha)$ is a vertex separator if there is no path from U_α to V_α in $\alpha \setminus S$. A minimum vertex separator S_{min} minimizes $|S_{\text{min}}|$ over all such separating sets.*

For $G_{n,1/2}$ with high probability the following norm bound holds for all shapes:

$$\|M_\alpha\| \leq \widetilde{\mathrm{O}}\left(n^{\frac{|V(\alpha)|-|S_{\text{min}}|}{2}}\right).$$

This achieves the correct polynomial magnitude in n (a matching lower bound up to log factors is proven in [AMP20, Appendix A]), but unfortunately, we lack a finer understanding of the behavior of $\|M_\alpha\|$. In Lemma 2.16 below, the log factor of $(\log n)^{\frac{1}{2}C_\alpha}$ is not optimal (for example, it is extraneous for the single-edge shape). The bulk distribution of the spectrum of singular values has been determined for certain "Z-matrices" by Cai and Potechin [CP20, CP22].

Two formal statements of the norm bound for $G_{n,1/2}$ are as follows. The first is stronger, and the second is easier to apply.

Lemma 2.16 ([AMP20, Theorem 6.1]). *For all (proper) shapes α and $\varepsilon > 0$, let*

$$\begin{aligned} C_\alpha &= |W_\alpha| + |S_{\text{min}}| - |U_\alpha \cap V_\alpha|, \\ D_\alpha &= |V(\alpha) \setminus (U_\alpha \cap V_\alpha)|, \end{aligned}$$

then provided that $C_\alpha > 0$:

$$\Pr\left[\|M_\alpha\| \geq \left(6e\left\lceil \frac{1}{3C_\alpha}\log(n^{|S_{\min}|}/\varepsilon)\right\rceil\right)^{\frac{1}{2}C_\alpha} \cdot (2D_\alpha)^{D_\alpha} \cdot n^{\frac{|V(\alpha)|-|S_{\min}|}{2}}\right] \leq \varepsilon\,.$$

Lemma 2.17 ([GJJ$^+$20, Appendix A]). *There is a universal constant C such that whp the following norm bound holds for all (proper) shapes α:*

$$\|M_\alpha\| \leq 2 \cdot (|V(\alpha)| \cdot \log(n))^{C \cdot |V(\alpha) \setminus (U_\alpha \cap V_\alpha)|} \cdot n^{\frac{|V(\alpha)|-|S_{\min}|}{2}}$$

The cited proofs use the "trace method", which we briefly review. Rajendran and Tulsiani [RT20] derive similar bounds using the matrix Efron-Stein inequality.

The trace method consists of the following steps to bound the spectral norm of a matrix M.

(i) Use $\|M\| \leq \operatorname{tr}((MM^\intercal)^q)^{1/2q}$ for any choice of $q \in \mathbb{N}$

(ii) $\Pr[\operatorname{tr}((MM^\intercal)^q)^{1/2q} \geq a] = \Pr[\operatorname{tr}((MM^\intercal)^q) \geq a^{2q}]$

(iii) Apply Markov's inequality: $\Pr[\operatorname{tr}((MM^\intercal)^q) \geq a^{2q}] \leq \frac{\mathbb{E}[\operatorname{tr}((MM^\intercal)^q)]}{a^{2q}}$

(iv) If the desired tail bound probability is ε, the tail is $a = \frac{\mathbb{E}[\operatorname{tr}((MM^\intercal)^q)]^{1/2q}}{\varepsilon^{1/2q}}$.

(v) Choose q.

The expected q-th trace can be expressed in a combinatorial way, reducing to (often challenging) combinatorial calculations. This is a combinatorial version of the "moment generating function" technique for tail bounds, which consists of the following steps to obtain an exponential tail bound on a random variable X.

(i) $\Pr[X \geq a] = \Pr[e^{tX} \geq e^{ta}]$ for any choice of t.

(ii) Apply Markov's inequality: $\Pr[e^{tX} \geq e^{ta}] \leq \frac{\mathbb{E}[e^{tX}]}{e^{ta}}$. (The expression $\mathbb{E}[e^{tX}]$ is the moment generating function of X evaluated at t.)

(iii) If the desired tail bound probability is ε, the tail is $a = \frac{\ln \mathbb{E}[e^{tX}] + \ln(1/\varepsilon)}{t}$.

(iv) Choose t.

Picking $t = q$ is a smooth way to approximate the q-th moment of X.

2.2.1 Sparse norm bounds

We will also work in the *sparse* setting. In the sparse regime, we work with matrices that are functions of an underlying random object $X \in \mathbb{R}^N$, but now the individual coordinates of X may have a distribution that depends on n.

The canonical setting is that X is an Erdős-Rényi random graph $G_{n,p}$ for $p = o(1)$. Each edge takes the p-biased distribution, and hence we use p-biased Fourier analysis. The definitions of ribbons, shapes, and graph matrices are essentially the same as in the case of $G_{n,1/2}$, with the sole change that the Fourier character $\chi_e(X)$ is replaced by the p-biased Fourier character (defined in Section 6.2.1 or [O'D14, Section 8.4]). The shapes and ribbons still correspond to graphs on $[n]$.

The norm bound of a graph matrix M_α is qualitatively different in $G_{n,p}$ vs $G_{n,1/2}$. It depends on a slightly modified definition of the minimum vertex separator. Let $E(S)$ be the edges induced by vertex set S. For $G_{n,p}$, the following norm bound holds for a shape α with high probability:

$$\|M_\alpha\| \leq \widetilde{\mathrm{O}}\left(\max_{\substack{\text{vertex separator } S \\ \text{of } \alpha}} \left\{ n^{\frac{|V(\alpha)|-|S|}{2}} \cdot \left(\frac{1-p}{p}\right)^{\frac{|E(S)|}{2}} \right\} \right).$$

A maximizer of the above is called a *sparse minimum vertex separator.*

Two formal statements are as follows.

Lemma 2.18 ([JPR+21, set $q = |V(\alpha)| \log n$ in Theorem 6.60]). *For a (proper) shape α, let:*

$$\begin{aligned} L_S &= \textit{vertices in } S \textit{ that are reachable from } U_\alpha \textit{ without passing through any other vertices in } S \\ R_S &= \textit{vertices in } S \textit{ that are reachable from } V_\alpha \textit{ without passing through any other vertices in } S \\ c(S) &= \textit{number of connected components of } \alpha \setminus S \textit{ that are not reachable from } U_\alpha \textit{ or } V_\alpha \end{aligned}$$

For all $\varepsilon > 0$, with probability at least $1 - \varepsilon$ the following holds:

$$\begin{aligned} \|M_\alpha\| \leq & \left(\frac{1}{\varepsilon}\right)^{\frac{1}{2|V(\alpha)|\log n}} \\ & \cdot 4^{|V(\alpha)| - \frac{|U_\alpha|+|V_\alpha|}{2}} \cdot \sqrt{|V(\alpha)|}^{|V(\alpha)\setminus(U_\alpha \cap V_\alpha)|} \\ & \max_{\substack{\text{vertex separator } S \\ \text{of } \alpha}} \left\{ n^{\frac{|V(\alpha)|-|S|}{2}} \left(3\sqrt{\frac{1-p}{p}}\right)^{|E(S)|} \sqrt{|V(\alpha)|}^{|S\setminus(U_\alpha\cap V_\alpha)|} (2|V(\alpha)|\log n)^{|S| - \frac{|L_S|+|R_S|}{2} + |c(S)|/2} \right\} \end{aligned}$$

where $\mathrm{Aut}(\alpha)$ *is defined in Section 2.3.2.*

Remark 2.19. *Unfortunately, this norm bound is not the "dream bound" for $G_{n,p}$ for at least two reasons.*

First, the bound is not tight up to $\mathrm{poly}(n)$ *factors in cases where the shape has too many edges. This is because the presence of a dense subgraph such as K_5 in the input can dramatically increase* *the norm of some graph matrices with many edges, and the norm bound must accommodate these rare events. With high probability K_5 is not present in the input, but K_5 is present with inverse* $\mathrm{poly}(n)$ *probability. Said in a different way, the stated bound holds (with* $\log n$ *factor loss) with very high probability (all but probability $n^{-\Omega(\log n)}$), whereas we are normally interested in what happens with high probability. To improve the norm bound for dense shapes in Chapter 6, we use an improved bound on the Frobenius norm.*

In fact, this is a fundamental difference between the sparse and dense settings. In the sparse setting, $\|M_\alpha\|$ is not exponentially concentrated around its mean. However, we believe that shapes without too many edges still exhibit exponential concentration.

The second defect with this bound is that the dependence on $|V(\alpha)|$ and $\log n$ *is sub-optimal. Removing all unnecessary factors is important for handling the regime when the random graph has average degree $o(\log n)$.*

Lemma 2.20. *There is a constant C such that with high probability, the following norm bound holds for all (proper) shapes α:*

$$\|M_\alpha\| \leq C^{|U_\alpha \cap V_\alpha|} \cdot (|V(\alpha)| \log n)^{C|V(\alpha) \setminus (U_\alpha \cap V_\alpha)|} \cdot \max_{\substack{\text{vertex separator } S \\ \text{of } \alpha}} n^{\frac{|V(\alpha)|-|S|}{2}} \left(3\sqrt{\frac{1-p}{p}}\right)^{|E(S)|}.$$

Proof. Let N_k be the number of shapes with k total vertices. Apply the formal norm bound Lemma 2.18 on shape α with parameter $1/\varepsilon = 1/\varepsilon(\alpha) = n \cdot N_{|V(\alpha)|} \cdot 2^{|V(\alpha)|}$, then use the union bound.

Summing over the shapes, the probability of failure is at most:

$$\begin{aligned} \sum_{\text{shape } \alpha} \varepsilon(\alpha) &= \sum_{k=1}^{n} N_k \cdot \frac{1}{nN_k 2^k} \\ &\leq \frac{1}{n}. \end{aligned}$$

Lemma 2.21. $N_k \leq 4^k (k!)^2 2^{k^2}$.

Proof. The following process forms all such shapes: starting from k vertices, decide whether each vertex is in U_α and/or V_α, then order U_α, V_α, then among the k^2 vertex pairs put any number of edges. □

It remains to simplify the bound, which is written below.

$$\begin{aligned}\|M_\alpha\| \leq & \left(\frac{1}{\varepsilon}\right)^{\frac{1}{2|V(\alpha)|\log n}}\\ & \cdot 4^{|V(\alpha)|-\frac{|U_\alpha|+|V_\alpha|}{2}} \cdot \sqrt{|V(\alpha)|}^{|V(\alpha)\setminus(U_\alpha\cap V_\alpha)|}\\ & \max_{\substack{\text{vertex separator } S\\ \text{of } \alpha}} \left\{ n^{\frac{|V(\alpha)|-|S|}{2}} \left(3\sqrt{\frac{1-p}{p}}\right)^{|E(S)|} \sqrt{|V(\alpha)|}^{|S\setminus(U_\alpha\cap V_\alpha)|} (2|V(\alpha)|\log n)^{|S|-\frac{|L_S|+|R_S|}{2}+|c(S)|/2} \right\}\end{aligned}$$

The first line is (using Lemma 2.21)

$$\begin{aligned}& \left(nN_{|V(\alpha)|}2^{|V(\alpha)|}\right)^{\frac{1}{2|V(\alpha)|\log n}}\\ \leq & \left(n4^{|V(\alpha)|}|V(\alpha)|^{2|V(\alpha)|}2^{|V(\alpha)|^2}2^{|V(\alpha)|}\right)^{\frac{1}{2|V(\alpha)|\log n}}\\ \leq & 4\cdot 2^{|V(\alpha)|}\\ = & 4\cdot 2^{|U_\alpha\cap V_\alpha|}\cdot 2^{|V(\alpha)\setminus(U_\alpha\cap V_\alpha)|}.\end{aligned}$$

The second line is

$$4^{|V(\alpha)|-\frac{|U_\alpha|+|V_\alpha|}{2}} \cdot \sqrt{|V(\alpha)|}^{|V(\alpha)\setminus(U_\alpha\cap V_\alpha)|} \leq (4\sqrt{|V(\alpha)|})^{|V(\alpha)\setminus(U_\alpha\cap V_\alpha)|}.$$

For the third line, we bound

$$\begin{aligned}& \sqrt{|V(\alpha)|}^{|S\setminus(U_\alpha\cap V_\alpha)|} \cdot (2|V(\alpha)|\log n)^{|S|-\frac{|L_S|+|R_S|}{2}+|c(S)|/2}\\ \leq & \sqrt{|V(\alpha)|}^{|V(\alpha)\setminus(U_\alpha\cap V_\alpha)|} \cdot (2|V(\alpha)|\log n)^{|V(\alpha)\setminus(U_\alpha\cap V_\alpha)|}.\end{aligned}$$

Multiplying the three bounds together yields the desired claim. □

2.3 Some technicalities and advice

The reality is that formally using graph matrices in proofs can be technically painful. Several concepts that we have encountered are described here, as well as what we found were the most effective ways of mitigating technical details.

In this section, we work with $G \sim G_{n,1/2}$ although the ideas apply more generally to other settings.

Contents of the section:

1. General tips

2. Automorphisms

3. Improper shapes

4. Ordered tuples vs sets for matrix indices

2.3.1 General tips

View M_α as a block matrix The index set $\mathcal{I}$ is all subsets of $[n]$. It is useful to partition the subsets by size and view matrices in $\mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ as block matrices. A graph matrix M_α lives in the $(|U_\alpha|, |V_\alpha|)$ block, and is zero on the other blocks. For example, this implies that the product $M_\alpha M_\beta$ is only nonzero if V_α and U_β "match up", $|V_\alpha| = |U_\beta|$.

First do combinatorial arguments using norm bounds Say we have a matrix written in the Fourier basis that we would like to prove something about, over a random input. For example, we may wish to prove a concentration inequality by bounding $\|\Lambda - \mathbb{E}\,\Lambda\|$ whp.

$$\Lambda = \sum_{\text{ribbons } R} \lambda_R M_R = \sum_{\text{shapes } \alpha} \lambda_\alpha M_\alpha\,.$$

The most important thing is to control norms of error terms at the coarse granularity of poly(n). To do this, one should think in the shape basis, using the norm bounds to obtain the magnitude in n of each shape (ignoring the subpolynomial factors in the norm bounds for the moment), and check that each individual term $|\lambda_\alpha|\|M_\alpha\|$ is small. Since $\|M_\alpha\|$ is determined by separators of α, this will require combinatorial arguments on the shapes to argue that the polynomial factors are under control.

Once the polynomial factors of n are managed, then we can check that the remaining $n^{o(1)}$ factors do not blow up (coming from subpolynomial factors in the norm bounds, λ_α, and counting arguments). We follow this approach in later chapters by first giving informal arguments that the powers of n are under control, and then moving on to the formal proofs.

Formally manipulate ribbons When it is time to formally manipulate an expression in the Fourier basis, it is usually easier to use ribbons. Each ribbon corresponds exactly with a combinatorial object on $[n]$ which can be manipulated while maintaining uniqueness of representation of the matrix. (Contrast this with embeddings in the next subsection, in which each ribbon is represented multiple times.) At the end of the argument, the ribbons are grouped into shapes, and we apply norm bounds. This strategy is used later in Section 6.4.1.

Here is an example of manipulating ribbons. When we have a matrix Λ written in the ribbon basis, often we can separate out $\Lambda[I, J]$ into a part that depends on the input restricted to $I \cup J$, and a part that depends on the rest of the input. Let R be a ribbon with no edges in $A_R \cup B_R$. We may group together the ribbons in Λ with R outside of $A_R \cup B_R$

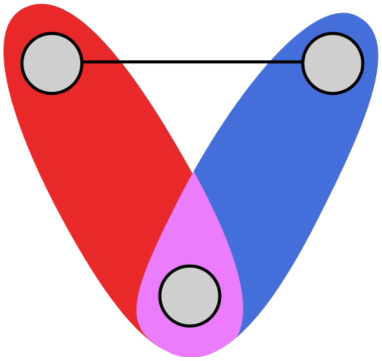


Figure 2.4: The purple vertex lies in $U_\alpha \cap V_\alpha$. For each choice of a label k for the purple vertex, for $i, j \in [n]$, the $((i,k),(j,k))$ entries of the graph matrix are essentially a copy of the one-edge shape with $U_\alpha \cap V_\alpha$ deleted. The $((i,k),(j,k))$ entry is $\chi_{\{i,j\}}(X)$ if i, j, k are distinct and 0 otherwise.

into the following summation:

$$\sum_{E \subseteq \binom{A_R \cup B_R}{2}} \lambda_{R \cup E} \cdot \chi_{E(R) \cup E} = \left(\sum_{E \subseteq \binom{A_R \cup B_R}{2}} \lambda_{R \cup E} \chi_E \right) \cdot \chi_{E(R)}$$

Assuming that the coefficients $\lambda_{R \cup E}$ factor into a part depending on E and a part depending on R, we may entirely factor out the dependence on R, so that the summation is now just a function of the input restricted to $A_R \cup B_R$. Doing this for example in Chapter 6, we will have:

$$\Lambda[I, J] = \mathbb{1}_{I \cup J \text{ is an independent set}} \cdot (\text{function independent of edges inside } I \cup J)\,.$$

Ignore $U_\alpha \cap V_\alpha$ Vertices inside $U_\alpha \cap V_\alpha$ can essentially be ignored. These vertices make M_α a block diagonal matrix. The "interesting" part of M_α is occurring on each of the blocks, and these blocks are close to being equal. For example, if α is the shape depicted in Fig. 2.4, then each block is essentially a copy of the adjacency matrix. Analysis of each block should be similar to the analysis when $U_\alpha \cap V_\alpha$ is deleted from the shape. It is prudent to first assume $U_\alpha \cap V_\alpha = \emptyset$, then handle the case $U_\alpha \cap V_\alpha \neq \emptyset$.[3]

[3] Warning: since the norm of the overall matrix is the maximum norm of the blocks, edges inside $U_\alpha \cap V_\alpha$ may contribute extra factors to the norm bound in the non-Boolean case. For example, if each edge of the input is a standard Gaussian random variable ($G_{i,j} \sim \mathcal{N}(0,1)$ instead of $G_{i,j} \in_R \{1, -1\}$) then a shape that consists of a single edge inside $U_\alpha \cap V_\alpha$ has $G_{s,t}$ in the diagonal entry $\{s,t\}$. The maximum of $G_{s,t}$ is of order $\Theta(\sqrt{\log n})$ whp, thus the norm of the matrix increases by a factor of order $\Theta(\sqrt{\log n})$.

2.3.2 Automorphisms

In order to simplify certain graph matrix calculations, it can be helpful to modify the definition of graph matrices given above. However, there is a tradeoff between simplifying the calculations and making the basic definitions more complex.

A graph matrix essentially sums over all embeddings of the shape α into the ambient graph G. Depending on the context, it is usually simpler to use this as the definition of a graph matrix.

Definition 2.22 (Alternate definition of graph matrix)**.** *For a shape α,*

$$M_\alpha = \sum_{\text{injective } \phi : V(\alpha) \to [n]} M_{\phi(\alpha)}\,.$$

This is not exactly equal to the sum over ribbons with shape α, since each ribbon arises from $|\mathrm{Aut}(\alpha)|$ different embeddings.

Definition 2.23 (Embedding)**.** *Given a shape α and an injective function $\phi : V(\alpha) \to [n]$, let $\phi(\alpha)$ be the ribbon obtained by labeling α in the natural way.*

Definition 2.24 (Automorphism)**.** *For a shape α, $\mathrm{Aut}(\alpha)$ is the stabilizer subgroup of any ribbon of shape α. Equivalently for $G_{n,1/2}$, $\mathrm{Aut}(\alpha)$ is the group of bijections from $V(\alpha)$ to itself such that U_α and V_α are fixed as sets and the map is a graph automorphism on $E(\alpha)$.*

Fact 2.25.

$$\sum_{\text{injective } \phi : V(\alpha) \to [n]} M_{\phi(\alpha)} = |\mathrm{Aut}(\alpha)| \sum_{R \text{ ribbon of shape } \alpha} M_R\,.$$

The sum over embeddings generally simplifies expressions. For example, the number of embeddings is just $n(n-1)\cdots(n-|V(\alpha)|+1)$, whereas the number of distinct ribbons of shape α is $\frac{n(n-1)\cdots(n-|V(\alpha)|+1)}{|\mathrm{Aut}(\alpha)|}$. Notice that the automorphism terms are present but "hiding" in the original definition. Automorphism terms $|\mathrm{Aut}(\alpha)|$ that appear in proofs need to be tracked and bounded, using propositions such as the following.

Proposition 2.26. *For a shape α, let $\alpha \pm e$ denote the shape with edge e added or deleted. Then*

$$\frac{|\mathrm{Aut}(\alpha \pm e)|}{|\mathrm{Aut}(\alpha)|} \;\le\; |V(\alpha)|^2.$$

Proof. We show that the two groups have a large subgroup which are equal. Consider $\mathrm{Aut}(\alpha \pm e)$ and $\mathrm{Aut}(\alpha)$ as group actions on the set $\binom{V(\alpha)}{2}$. Letting G^e denote the stabilizer of edge e, observe that $\mathrm{Aut}(\alpha \pm e)^e = \mathrm{Aut}(\alpha)^e$. By the orbit-stabilizer lemma, the index $|G : G^e|$ is equal to the size of the orbit of e, which is at least 1 and at most $|V(\alpha)|^2$. So,

$$\frac{|\mathrm{Aut}(\alpha \pm e)|}{|\mathrm{Aut}(\alpha)|} = \frac{|\mathrm{Aut}(\alpha \pm e) : \mathrm{Aut}(\alpha \pm e)^e|}{|\mathrm{Aut}(\alpha) : \mathrm{Aut}(\alpha)^e|} \le |V(\alpha)|^2. \qquad \square$$

2.3.3 Improper shapes and linearization

This trick is generally applicable when performing Fourier analysis. Assume we have a function f expressed in the Fourier basis, and one step of the proof is to multiply f with another function g also expressed in the Fourier basis,

$$f \cdot g = \sum_{\alpha,\beta} \widehat{f}(\alpha)\widehat{g}(\alpha)\chi_\alpha \chi_\beta \,.$$

To re-express this in the Fourier basis, we would need to *linearize*

$$\chi_\alpha \chi_\beta = \sum_{\gamma} c_\gamma^{\alpha,\beta} \chi_\gamma \,.$$

For example, using the Hermite polynomials $h_i(x)$ (the orthogonal polynomials for the standard Gaussian distribution $\mathcal{N}(0,1)$), an example linearization is:

$$h_1(x)^2 = x^2 = (x^2 - 1) + 1 = h_2(x) + h_0(x).$$

Due to orthogonality of the Fourier basis, the coefficients are $c_\gamma^{\alpha,\beta} = \frac{\mathbb{E}[\chi_\alpha \chi_\beta \chi_\gamma]}{\mathbb{E}[\chi_\gamma^2]}$. However, if we still have more multiplications to do, we can delay the linearization until the end of the proof. We allow f to be expressed as a linear combination of "improper" Fourier basis functions $\chi_{\alpha_1}\chi_{\alpha_2}\cdots\chi_{\alpha_k}$ during intermediate calculations.

In the context of $G_{n,1/2}$, improper Fourier characters are multigraphs instead of graphs. Parallel multiedges equal the product of Fourier characters. We also allow ribbons or shapes to be improper in a second way, by having isolated vertices.

Definition 2.27 (Improper ribbon for $G_{n,1/2}$)**.** *A ribbon is a tuple $R = (A_R, B_R, V(R), E(R))$ where $A_R, B_R \subseteq V(R) \subseteq [n]$ and $E(R)$ is a multigraph on $V(R)$. The corresponding matrix $M_R \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ is:*

$$M_R[I,J] = \begin{cases} \prod\limits_{e \in E(R)} \chi_e & I = A_R, J = B_R \\ 0 & \textit{otherwise}\,. \end{cases}$$

A shape is defined as before to be an equivalence class of ribbons under relabeling.

Definition 2.28 (Isolated vertex)**.** *For a shape α, an isolated vertex is a degree-0 vertex in $W_\alpha = V(\alpha) \setminus (U_\alpha \cup V_\alpha)$. Let I_α denote the set of isolated vertices in α. Similarly, for a ribbon R, the isolated vertices are denoted I_R.*

We stress that an isolated vertex never refers to degree-0 vertices inside $U_\alpha \cup V_\alpha$.

Definition 2.29 (Proper)**.** *A ribbon or shape is proper if it has no multi-edges and no isolated vertices. Otherwise, it is improper.*

An improper ribbon or shape can be decomposed in a unique way into a linear combination of proper ones, which we call *linearizations*.

Definition 2.30 (Linearization)**.** *Given an improper ribbon or shape α, a linearization β is a proper ribbon or shape such that M_β has nonzero coefficient in the Fourier expansion of M_α.*

For ribbons or graph matrices defined as a sum over embeddings, Definition 2.22, the coefficient is obtained by linearizing each edge separately (when using Definition 2.3 there is additionally an automorphism term). Linearization is particularly easy in the case of $G_{n,1/2}$. Since $\chi_e^2 = 1$, shape α is linearized by taking the multiplicity of each edge mod 2.

Improper shapes with isolated vertices are scalar multiples of the shape with the isolated vertices removed. Adding an isolated vertex to a shape α for $G_{n,1/2}$ multiplies the number of embeddings by $n - |V(\alpha)|$.

To get norm bounds for improper shapes, we linearize the shape and take the largest norm bound among the resulting proper shapes. The usual norm bound states that each non-isolated vertex outside the minimum vertex separator gives a factor of $\sqrt{n}$; now additionally, each isolated vertex contributes a factor of n.

In the dense setting, removing edges from a shape can only increase the norm. The size of the minimum vertex separator may decrease (increasing the norm) or the number of isolated vertices may increase (increasing the norm). In the sparse setting, removing edges inside a sparse minimum vertex separator may decrease the norm.

2.3.4 Unordered sets for matrix indices

In later chapters, it will almost always suffice to use matrices with left/right sides indexed by *unordered* sets (this aligns more closely with the sum-of-squares algorithm). To use this definition for $G_{n,1/2}$, we take $\mathcal{I}$ to be the collection of all subsets of $[n]$, and ribbons and shapes are defined as before.

We view this definition as slightly less canonical for the reason that some matrix entries are the sum of more than one Fourier character. For example, consider the shape for $G_{n,1/2}$ depicted in Fig. 2.5.

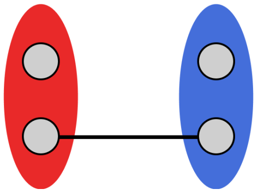


Figure 2.5

If unordered sets are used, then in entry $(\{i,j\},\{k,\ell\})$ there are the four Fourier characters

$$\chi_{\{i,k\}} + \chi_{\{j,k\}} + \chi_{\{i,\ell\}} + \chi_{\{j,\ell\}}\,.$$

If ordered sets are used (say that the upper vertex in U_α and V_α comes first in the order), then in entry $((i,j),(k,\ell))$ there is only the Fourier character $\chi_{\{j,\ell\}}$. When using ordered sets, if $W_\alpha = \emptyset$, then each nonzero entry of M_α contains exactly one Fourier character.

When using ordered sets, there is exactly one way to factor a graph matrix across a vertex separator, and exactly one "dominant" shape when two graph matrices are multiplied. This is helpful when using the trace method or in other places where graph matrices are multiplied or factored. See the next section for more details. From a practical point of view, sometimes this extra level of control is not needed.

2.4 Multiplying and factoring graph matrices

One of the key facts about graph matrices is that multiplication of graph matrices approximately equals a new graph matrix, $M_\alpha \cdot M_\beta \approx M_\gamma$, where γ is the result of "gluing" V_α with U_β (and if V_α, U_β cannot be glued because they do not have the same number of vertices, the product is zero). The unaccounted terms in the approximation are *intersection terms* between α and β.

Example 2.31 (Squared Wigner matrix)**.** *Let A be the symmetric $n \times n$ Wigner matrix with ± 1 entries and 0 on the diagonal from Example 2.14. The spectrum of the matrix $AA = A^\intercal A$ is the square of the Wigner semicircle distribution, and hence it is supported on $[0, (4 + o(1))n]$ with high probability. The entries of AA are given by:*

$$(AA)_{i,j} = \sum_{k=1}^{n} A_{i,k}A_{j,k}\,, \qquad\qquad (AA)_{i,i} = \sum_{k=1}^{n} A_{i,k}A_{i,k} = n - 1\,.$$

This calculation may also be done diagrammatically. A is represented by the graph matrix for the one-edge shape, and AA is computed in Fig. 2.6, using improper shapes (Section 2.3.3) on the way.

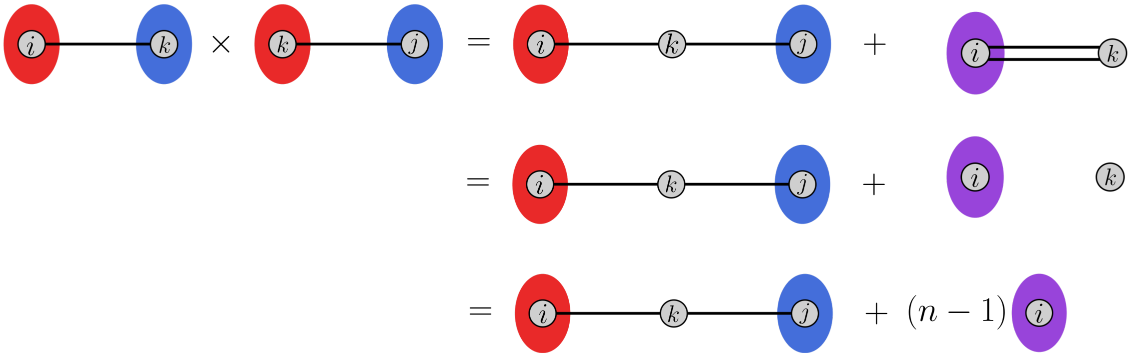


Figure 2.6: (First equality) The left side sums over all i, j, k. The first term on the right side has terms where $i \neq j$, and the second term has terms where $i = j$ (purple denotes $U_\alpha \cap V_\alpha$). The second term is an improper shape, since when $i = j$, both edges become $\{i, k\}$. (Second equality) Since A has Boolean entries, linearize $A_{i,k}^2 = 1$. (Third equality) The isolated vertex scales the matrix by $n - 1$.

The 2-path is the "main" term created by gluing two copies of the one-edge shape, while the *singleton purple vertex is an intersection term (a multiple of the identity matrix). By subtracting* $(n-1)\cdot \mathrm{Id}$ *from both sides, we see that the spectrum of the 2-path is* $[-n, (3+o(1))n]$ *whp.*

In this example, the norm of the intersection term $(n-1)\cdot\mathrm{Id}$ *is of the same order as the "main" term, namely* $\Theta(n)$. *In other cases where* α *and* β *are "left" and "right" shapes, the intersection terms between* $M_\alpha M_\beta$ *will have smaller norm than the "main" term.*

Remark 2.32. *Recall the tip from the previous section to formally manipulate ribbons instead of graph matrices. When following this advice, formally we will not multiply or factor graph matrices themselves, only ribbons. However, the ribbon analysis is performed with the intuition of graph matrices in mind.*

Definition 2.33 (Composable). *Ribbons* R, S *are composable if* $B_R = A_S$. *Shapes* α, β *are composable if* V_α *and* U_β *specify the same subset of* $\mathcal{I}$.

Multiplication of ribbons exactly equals a new ribbon, although it may be improper.

Definition 2.34 (Composing ribbons). *The composition* $R \circ S$ *of composable ribbons* R, S *is the (possibly improper) ribbon* $T = (A_R, B_S, E(R) \sqcup E(S))$.

Fact 2.35. *If* R, S *are composable ribbons, then* $M_{R\circ S} = M_R M_S$.

Definition 2.36 (Composing shapes). *The composition* $\alpha \circ \beta$ *of composable shapes* α, β *is the (possibly improper) shape* ζ *whose multigraph is the result of gluing together the graphs for* α, β *along* V_α *and* U_β. *Set* $U_\zeta = U_\alpha$ *and* $V_\zeta = V_\beta$.

If we write $\alpha \circ \beta$ then we will implicitly assume that α and β are composable. We would like to say that the graph matrix $M_{\alpha\circ\beta}$ also factors as $M_\alpha M_\beta$. This is approximately but not exactly true; there are *intersection terms*.

Definition 2.37 (Intersection pattern). *For composable shapes* $\alpha_1, \alpha_2, \dots, \alpha_k$, *let* $\alpha = \alpha_1 \circ \alpha_2 \circ \cdots \circ \alpha_k$. *An intersection pattern* P *is a partition of* $V(\alpha)$ *such that for all* i *and* $v, w \in V(\alpha_i)$, v *and* w *are not in the same block of the partition. We say that a vertex "intersects" if its block has size at least 2 and let* $\mathrm{Int}(P)$ *denote the set of intersecting vertices.*

Let $\mathcal{P}_{\alpha_1,\alpha_2,\dots,\alpha_k}$ *be the set of intersection patterns between* $\alpha_1, \alpha_2, \dots, \alpha_k$.

Definition 2.38 (Intersection shape). *For composable shapes* $\alpha_1, \alpha_2, \dots, \alpha_k$ *and an intersection pattern* $P \in \mathcal{P}_{\alpha_1,\alpha_2,\dots,\alpha_k}$, *let* $\alpha_P = \alpha_1 \circ \alpha_2 \circ \cdots \circ \alpha_k$ *then identify all vertices in blocks of* P, *i.e. contract them into a single super vertex. Keep all edges (and hence* α_P *may be improper).*

Composable ribbons $R_1, \dots, R_k$ with shapes $\alpha_1, \dots, \alpha_k$ induce an intersection pattern $P \in \mathcal{P}_{\alpha_1,\dots,\alpha_k}$ based on which vertices are equal. When multiplying graph matrices defined using the embedding definition (Definition 2.22), by casing on which vertices are equal we have:

Proposition 2.39. *For composable shapes* $\alpha_1, \alpha_2, \dots, \alpha_k$,

$$M_{\alpha_1} \cdots M_{\alpha_k} = \sum_{P \in \mathcal{P}_{\alpha_1,\dots,\alpha_k}} M_{\alpha_P}.$$

Note that different P may give the same α_P, and hence the total coefficient on M_{α_P} for a given shape α_P may be complicated. In practice the best way to determine the coefficient of α_P is to fix any ribbon of shape α_P on the right-hand side and sum (or upper bound) the possible ribbons on the left-hand side that contribute to it.

We think of the "main terms" as being those where no vertices intersect, i.e. the ribbons have no vertices in common except on the boundaries between consecutive ribbons. "Main" intersection patterns glue together $\alpha_1 \circ \cdots \circ \alpha_k$ but do not contract any vertices.

The terms in which at least one vertex intersects are the intersection terms. In general, the norm of an intersection term could be larger than the main terms. However, the norms are smaller if the factorization has a special type, based on vertex separators.[4]

Definition 2.40 (Leftmost/rightmost minimum vertex separators)**.** *For a ribbon or shape α, a vertex separator S is the leftmost minimum vertex separator (LMVS) if $\alpha \setminus S$ has no path from U_α to any other minimum vertex separator. The rightmost minimum vertex separator (RMVS) likewise cuts paths from V_α.*

The LMVS and RMVS can be shown to be uniquely defined (the same proof works for either the dense MVS or the sparse MVS).

Proposition 2.41. *The LMVS is uniquely defined.*

Proof. Let S_1, S_2 be two minimum vertex separators. Then we can construct a minimum vertex separator to the left of both of them as follows (see Fig. 2.7 for a picture). Since this process cannot continue indefinitely, it must terminate in the LMVS.

Let $L_1 \subseteq S_1$ be vertices of S_1 reachable from U_α without passing through S_2, and likewise for $L_2 \subseteq S_2$. Then we take $S_L := L_1 \cup L_2 \cup (S_1 \cap S_2)$.

To show that S_L is a vertex separator, take a path P from U_α. Without loss of generality, P passes through S_1 before S_2 (or at the same time). Then L_1 (or $S_1 \cap S_2$) blocks P.

To show that S_L is minimum, observe that if we perform the analogous construction of S_R, then $S_L \sqcup S_R = S_1 \sqcup S_2$. If one of S_L, S_R is larger, then the other must be smaller. Since S_1, S_2 are both minimum, neither S_L nor S_R can be smaller, and we conclude that both S_L, S_R are in fact minimum vertex separators. □

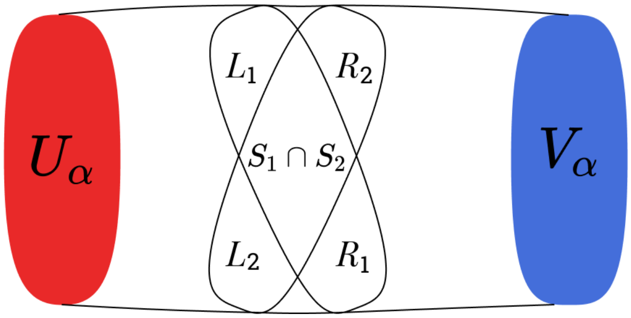


Figure 2.7

[4] For dense inputs, this is the *intersection tradeoff lemma*, Lemma 6.25. For sparse inputs, this remains conjectural.

Definition 2.42 (Left shape). *A shape σ is a left shape if it is proper, the unique minimum vertex separator is V_σ, there are no edges with both endpoints in V_σ, and every vertex is connected to U_σ.*

Definition 2.43 (Middle shape). *A shape τ is a middle shape if U_τ is the leftmost minimum vertex separator of τ, and V_τ is the rightmost minimum vertex separator of τ. If τ is proper, we say it is a proper middle shape.*

Definition 2.44 (Right shape). *σ' is a right shape if $\sigma'^\intercal$ is a left shape.*

We also extend the definition of (L/R)MVS, left, middle, and right to ribbons. Every proper ribbon or shape admits a canonical decomposition into left, right, and middle parts.

Proposition 2.45. *Every proper ribbon or shape α has a unique decomposition $\alpha = \sigma \circ \tau \circ \sigma'^\intercal$, where σ is a left shape, τ is a middle shape, and $\sigma'^\intercal$ is a right shape.*

Proof. The decomposition takes σ to be the set of vertices reachable from U_α via paths that do not pass through the LMVS, similarly for σ' vis-à-vis the RMVS, and then τ is the remainder. □

2.5 Open Problems

The first open problem is to improve our understanding of the spectrum of graph matrices, for a random input.

(1) What is the correct norm of M_α, up to logarithmic or constant factors, in either the sparse or dense case [AMP20]?

(2) What is the distribution of the *spectral edge*, i.e. the fluctuation of the top eigenvalue?

(3) What is the shape of the *spectral bulk* [CP20, CP22]?

In the case of a Wigner matrix, these three things are respectively $2\sqrt{n}$, the Tracy-Widom law, and the Wigner semicircle law.

The norm bounds on graph matrices that we cite are proven by combinatorially analyzing the moments $\mathbb{E}[\operatorname{tr}(M_\alpha M_\alpha^\intercal)^q]$ for $q \approx \log n$. On the one hand, strengthening the combinatorial analysis of the q-th moment would let us improve (1) and (3). On the other hand, if we just want (1) norm bounds on $\|M_\alpha\|$ up to constants, then an exponential tail bound would be good enough, and this may be easier to prove. To illustrate, suppose we could prove that, for some number $B > 0$, there exists a constant $t > 0$ such that for all unit vectors x:

$$\mathop{\mathbb{E}}_{G \sim G_{n,1/2}} \left[e^{x^\intercal M_\alpha^\intercal M_\alpha x t} \right] \le e^B .$$

Then for fixed x, the probability that the Rayleigh quotient $x^\intercal M_\alpha^\intercal M_\alpha x$ exceeds $O(B)$ exhibits exponential decay:

$$\begin{aligned}\Pr[x^\intercal M_\alpha^\intercal M_\alpha x \geq 10B/t] &= \Pr[e^{x^\intercal M_\alpha^\intercal M_\alpha x t} \geq e^{10B}] \\ &\leq \frac{\mathbb{E}_{G\sim G_{n,1/2}}[e^{x^\intercal M_\alpha^\intercal M_\alpha x t}]}{e^{10B}} \qquad \text{(Markov's inequality)} \\ &\leq \frac{1}{e^{9B}}\,.\end{aligned}$$

Because of the exponential decay, we may union bound over an epsilon net of unit vectors x, which has exponential size but with a smaller base. Passing from the epsilon net to the unit sphere, the same qualitative norm bound $O(B)$ holds for all x with all exponentially small error probability.

The theory of graph matrices uses an orthogonal Fourier basis on the random input. We don't know of a good way to generalize this to certain inputs that are of interest, and have non-independent entries, such as a random d-regular graph or a random unit vector. The proofs in Chapter 4 and Chapter 6 will boil down to graph matrix analysis. If the theory could be generalized to these other distributions, it is likely that those proofs would also apply.

Unlike the dense minimum vertex separator (which can be computed via min cut/max flow), we don't know an efficient algorithm for computing the sparse minimum vertex separator of a shape. We conjecture that this task is NP-hard.

Conjecture 2.46. *For all $C \in (0,1)$ the following problem is NP-hard: given an undirected graph G with two specialized vertex sets $A, B \subseteq V(G)$, compute*

$$\min_{S\text{: vertex separator of } A,B} |V(S)| - C \cdot |E(S)|\,.$$

CHAPTER 3
OVERVIEW OF SUM-OF-SQUARES LOWER BOUNDS

The *sum-of-squares (SoS) hierarchy* (also known as the *Lasserre hierarchy*) is a semidefinite programming hierarchy which provides a meta-algorithm for polynomial optimization. Due to the versatility of polynomials in modeling computational problems, the SoS hierarchy can be applied to a vast range of optimization and recovery problems. We study *combinatorial optimization problems*, in which we are looking for the best object among some finite collection (such as the best partition of a set into groups, in comparison to e.g. "which pair of points on the land surface of the Earth are as far apart as possible by land travel only?").

SoS has been shown to be quite successful at combinatorial optimization. It captures the best known approximation algorithms for several classical combinatorial optimization problems. Additional successes of SoS include Tensor PCA [HSS15, MSS16], Constraint Satisfaction Problems with additional structure [AJT19], recent breakthroughs in robust statistics [HL18, RSS18, KKM18, BP21], and our best algorithms for Unique Games [BRS11, GS11, BBK$^+$21a, BHKL22].

Here we are interested in proving lower bounds against sum-of-squares: when does it fail to solve a task? An SoS lower bound can serve as strong evidence for computational hardness of a given problem. This hardness evidence is particularly relevant to average-case problems, where relatively few techniques exist for establishing NP-hardness results [BABB21]. Furthermore, an SoS lower bound is *unconditional*: even if the Unique Games Conjecture is false, for example, a lower bound still proves that sum-of-squares techniques fail. Since SoS is a proof system capturing a class of algorithmic reasoning, the lower bound informs the algorithm designer to avoid methods of proof that are captured by low-degree SoS reasoning.

In this expository chapter we define the SoS algorithm and introduce our lower bound techniques. Section 3.1 defines the Sum-of-Squares algorithm and the related sum-of-squares proof system. The lower bounds proven in later chapters have two major conceptual steps: *pseudocalibration* (Section 3.2) and *graph matrix analysis* (Section 3.3).

Bibliography. The exposition in this chapter is extended from our works [GJJ$^+$20, JPR$^+$21]. The lower bound technique of pseudocalibration plus graph matrices was introduced by Barak et al in the context of the Planted Clique problem [BHK$^+$16], and laid out in fairly large generality by Potechin and Rajendran [PR20]. For more introduction to sum-of-squares, see [BS16, FKP19].

We regretfully omit important design considerations for sum-of-squares upper bounds (viz. algorithms), as well as other lower bound techniques such as [MRX20, Kun20].

3.1 The sum-of-squares algorithm

Given a system of polynomial equalities and inequalities $\{f_i(\mathsf{X}_1, \ldots, \mathsf{X}_n) = 0, g_j(\mathsf{X}_1, \ldots, \mathsf{X}_n) \geq 0\}$, the sum-of-squares (SoS) algorithm attempts to refute the existence of a feasible point

$\mathsf{X} \in \mathbb{R}^n$ satisfying all of the given constraints. The output of the algorithm is either "infeasible" or "may be feasible". This may also be used to approximate $\max p(\mathsf{X}_1, \dots, \mathsf{X}_n)$ for a polynomial p subject to polynomial constraints by binary searching for the smallest $C \in \mathbb{R}$ such that SoS can refute "$p(\mathsf{X}_1 \dots, \mathsf{X}_n) \geq C$".

The SoS framework specifies a *hierarchy* of programs, where each program is a convex relaxation of the polynomial optimization problem. This hierarchy is indexed by a parameter D called the *SoS degree*. By taking larger D, one gets a stronger algorithm (i.e., an algorithm that is able to refute additional infeasible systems) at the expense of a larger program with a slower runtime. The degree-D SoS algorithm can be implemented in time $n^{O(D)}$ via semidefinite programming (ignoring some issues of bit complexity [RW17]). $D = O(1)$ corresponds to polynomial time while $D = 2n$ is able to exactly solve an optimization problem on n Boolean variables in exponential time. Thus we are interested in the tradeoff between degree and performance of the algorithm. We informally say that SoS "fails to solve a problem" if the degree required to refute or optimize the system is $\omega(1)$, as in this case we have ruled out polynomial time SoS.

SoS attempts to refute a polynomial system by constructing a "degree-D sum-of-squares proof" that the system is infeasible. We will almost entirely use the dual viewpoint of SoS, through *pseudoexpectation operators* (the primal viewpoint of "degree-D sum-of-squares proof" is described briefly in Section 3.1.1).[1]

The pseudoexpectation operator specifies a "fake distribution of solutions" to the polynomial system. The SoS algorithm rules out the existence of a feasible point by checking for the existence of a fake distribution of solutions; if no such fake distribution exists, the system is guaranteed to be infeasible. Notice that this is a relaxation: if a fake distribution exists, it may or may not correspond to a true distribution of feasible points. To prove a lower bound against SoS, we will construct a fake distribution of solutions (viz. a valid pseudoexpectation operator) in situations where the system is truly infeasible.

Moving to formal definitions, let $\mathbb{R}^{\leq D}[\mathsf{X}_1, \dots, \mathsf{X}_n]$ be the set of polynomials of degree at most D in variables $\mathsf{X}_1, \dots, \mathsf{X}_n$. We denote the degree of a polynomial f by $\deg(f)$.

Definition 3.1 (Pseudoexpectation)**.** *Given a set of variables $\mathsf{X}_1, \dots, \mathsf{X}_n$, a degree-$D$ pseudoexpectation operator is a linear functional $\widetilde{\mathbb{E}} : \mathbb{R}^{\leq D}[\mathsf{X}_1, \dots, \mathsf{X}_n] \to \mathbb{R}$ such that $\widetilde{\mathbb{E}}[1] = 1$.*

The "fake distribution" specified by $\widetilde{\mathbb{E}}$ is in an implicit form: we are not given access to the probability density function, we are only given access to the expectation of the distribution on low-degree polynomials $\widetilde{\mathbb{E}}[p(\mathsf{X})]$. Observe that since $\widetilde{\mathbb{E}}$ is a linear function, it can be specified by its value on monomials up to degree D i.e. $\widetilde{\mathbb{E}}$ specifies the low-degree moments of the fake distribution.

The fake distribution needs to look feasible.

Definition 3.2 (Satisfying a constraint)**.** *A degree-D pseudoexpectation operator $\widetilde{\mathbb{E}}$ satisfies a*

[1] The duality may be viewed as *convex duality* of the convex relaxation. It morally underpins the analysis of SoS: the performance of the SoS algorithm can be upper and lower bounded by exhibiting a feasible primal solution or a feasible dual solution, and therefore "all we need to do" to pinpoint the performance of SoS is construct these concrete mathematical objects.

polynomial constraint "$f(\mathsf{X}) = 0$" if $\widetilde{\mathbb{E}}[f(\mathsf{X})p(\mathsf{X})] = 0$ for all $p(\mathsf{X})$ such that $\deg(p) + \deg(f) \leq D$. *Similarly, $\widetilde{\mathbb{E}}$ satisfies constraint "$g(\mathsf{X}) \geq 0$" if $\widetilde{\mathbb{E}}[g(\mathsf{X})p(\mathsf{X})^2] \geq 0$ for all $p(\mathsf{X})$ such that* $2\deg(p) + \deg(f) \leq D$.

Remark 3.3. *When the problem has Boolean constraints "$\mathsf{X}_i^2 = 1$" or "$\mathsf{X}_i^2 = \mathsf{X}_i$", any $\widetilde{\mathbb{E}}$ that satisfies these constraints is defined by its value on just multilinear monomials.*

The fake distribution is also required to be nonnegative on square polynomials. This is the defining property of the SoS algorithm and it is what gives the algorithm its strength, as it nontrivially restricts the space of pseudoexpectation operators.

Definition 3.4 (SoS-feasible)**.** *A degree-D pseudoexpectation operator $\widetilde{\mathbb{E}}$ is SoS-feasible if for every polynomial $p \in \mathbb{R}^{\leq D/2}[\mathsf{X}_1, \dots, \mathsf{X}_n]$, $\widetilde{\mathbb{E}}[p(\mathsf{X})^2] \geq 0$.*

Remark 3.5. *This is equivalent to satisfying the constraint "$1 \geq 0$".*

Definition 3.6 (Sum-of-squares algorithm)**.** *Given a system of polynomial constraints $\{f_i(\mathsf{X}) = 0, g_j(\mathsf{X}) \geq 0\}$ in n variables $\mathsf{X}_1, \dots, \mathsf{X}_n$, the degree-$D$ Sum-of-Squares algorithm checks for the existence of an SoS-feasible degree-D pseudoexpectation operator $\widetilde{\mathbb{E}}$ that satisfies the constraints. If $\widetilde{\mathbb{E}}$ exists, the algorithm outputs "may be feasible", otherwise it outputs "infeasible".*

In other words, degree-D SoS checks feasibility of the following program:

Variable: $\widetilde{\mathbb{E}} : \mathbb{R}^{\leq D}[\mathsf{X}_1, \dots, \mathsf{X}_n] \to \mathbb{R}$	linear	
$\widetilde{\mathbb{E}}[1] = 1$		(Normalization)
$\widetilde{\mathbb{E}}[f_i(\mathsf{X})p(\mathsf{X})] = 0$	$\forall i.\ \forall \deg(p) \leq D - \deg(f_i)$	(Given equalities)
$\widetilde{\mathbb{E}}[g_j(\mathsf{X})p(\mathsf{X})^2] \geq 0$	$\forall j.\ \forall \deg(p) \leq \dfrac{D - \deg(g_j)}{2}$	(Given inequalities)
$\widetilde{\mathbb{E}}[p(\mathsf{X})^2] \geq 0$	$\forall \deg(p) \leq \dfrac{D}{2}$	(Sum-of-squares)

The SoS algorithm can be implemented by a size $n^{O(D)}$ semidefinite program (SDP), as we now describe. In practice we will work with the SDP formulation of SoS so that we may use linear algebraic methods. The pseudoexpectation operator is equivalent to a *moment matrix*.

Assume D is even.

Definition 3.7 (Moment Matrix of $\widetilde{\mathbb{E}}$)**.** *The moment matrix $\mathcal{M} = \mathcal{M}(\widetilde{\mathbb{E}})$ associated to a degree-D pseudoexpectation $\widetilde{\mathbb{E}}$ is a $\left(\!\binom{[n]}{\leq D/2}\!\right) \times \left(\!\binom{[n]}{\leq D/2}\!\right)$ matrix with rows and columns indexed by multisets $I, J \subseteq [n]$ of size at most $D/2$ and defined as*

$$\mathcal{M}[I, J] := \widetilde{\mathbb{E}}\left[\mathsf{X}^I \cdot \mathsf{X}^J\right].$$

The dimension of the moment matrix is $\left(\!\binom{n}{\leq D/2}\!\right) = n^{O(D)}$. The moment matrix is naturally a block matrix, with blocks corresponding to the degree of the row and column monomials.

Definition 3.8 (Block). *For $k, l \in \mathbb{N}$, the (k, l) block of $\mathcal{M}$ is the submatrix with rows from $\left(\!\!\binom{[n]}{k}\!\!\right)$ and columns from $\left(\!\!\binom{[n]}{l}\!\!\right)$.*

The entries of the moment matrix only depend on the union of the row and column indices.

Definition 3.9 (SoS-symmetric). *A matrix $\mathcal{M}$ with rows and columns indexed by multisets from $[n]$ is SoS-symmetric if $\mathcal{M}[I, J] = \mathcal{M}[I', J']$ whenever $I \sqcup J = I' \sqcup J'$.*

Remark 3.10. *When a problem has Boolean variables, it suffices to use $\mathcal{M}$ with rows and columns indexed by subsets of size at most $D/2$ instead of multisets, as in Remark 3.3. Furthermore, $\mathcal{M}[I, J] = \widetilde{\mathbb{E}}[\mathsf{X}^I \mathsf{X}^J]$ depends only on $I \triangle J$ (in the presence of Boolean constraints "$\mathsf{X}_i^2 = 1$") or $I \cup J$ (Boolean constraints "$\mathsf{X}_i^2 = \mathsf{X}_i$").*

The crucial SoS-feasibility of the pseudoexpectation operator is equivalent to positive semidefiniteness (PSD-ness) of the moment matrix.

Fact 3.11. *The condition $\widetilde{\mathbb{E}}[f^2] \geq 0$ for all $\deg(f) \leq D/2$ is equivalent to $\mathcal{M}(\widetilde{\mathbb{E}}) \succeq 0$.*

With these constraints in place, moment matrices are equivalent to pseudoexpectation operators. The SDP formulation of SoS checks for the existence of a moment matrix. Recall that a semidefinite program is a linear program where the variables form a matrix that additionally must be PSD. Degree-D SoS is equivalent to feasibility of the following SDP. (For simplicity we only show the case without inequality constraints. See [FKP19, Section 3.2.1] for inequality constraints.)

Definition 3.12 (Sum-of-squares SDP). *Let $\langle A, B \rangle$ denote the entrywise (Frobenius) dot product of matrices. Let M_p be a matrix that encodes a polynomial $p(\mathsf{X})$: for each nonzero monomial $c_\alpha \mathsf{X}^\alpha$ in p, choose exactly one I, J with $I \sqcup J = \alpha$ and set $M_p[I, J] = c_\alpha$.*

Given a system of polynomial constraints $\{f_i(\mathsf{X}) = 0\}$ in n variables $\mathsf{X}_1, \ldots, \mathsf{X}_n$, the degree-$D$ sum-of-squares semidefinite program (SDP) is:

$$\boxed{\begin{array}{llll}
\quad\text{Variable:} & M \in \mathbb{R}^{\left(\!\binom{[n]}{\leq D/2}\!\right) \times \left(\!\binom{[n]}{\leq D/2}\!\right)} \text{ symmetric} & & \\
\text{s.t.} & & & \\
\quad M[\emptyset, \emptyset] = 1 & & & \text{(Normalization)} \\
\quad \left\langle M, M_{\mathsf{X}^\alpha f_i} \right\rangle = 0 & \forall i.\ \forall \alpha \in \left(\!\!\binom{[n]}{\leq D - \deg(f_i)}\!\!\right) & & \text{(Given equalities)} \\
\quad M[I, J] = M[I', J'] & \forall I \sqcup J = I' \sqcup J' & & \text{(SoS symmetry)} \\
\quad M \succeq 0 & & & \text{(PSD-ness)}
\end{array}}$$

Proposition 3.13. *For all even D, the existence of a degree-D pseudoexpectation operator is equivalent to feasibility of the degree-D sum-of-squares SDP.*

3.1.1 Sum-of-squares proofs

Sum-of-squares algorithmatizes a particular type of reasoning, known as *sum-of-squares reasoning*. Given a polynomial system, the degree-D SoS algorithm attempts to prove that the system is infeasible using only *non-negativity of squared polynomials* and *degree-$D/2$ local reasoning* (local reasoning is implied by non-negativity of squares, but for intuition we prefer to think of them as separate).

SoS is naturally a *certification* algorithm, rather than a search algorithm. That is, it solves the algorithmic task where we ask for an efficient proof that some structure does not exist in the input (for example, prove to me that the input graph does not have a k-clique). This is arguably a less natural task than search. For typical search tasks which are in NP (such as k-clique), because we conjecture $\mathsf{NP} \neq \mathsf{co\text{-}NP}$, we suspect that in general such efficient proofs don't even exist, independent of whether we can algorithmically construct them.

We are usually interested in specific cases of particular inputs, for example whether or not SoS (or any polynomial time algorithm) can disprove the existence of a large clique in a random graph. When used for optimization, certification algorithms can be seen as upper bounding the optimum value OPT, whereas search algorithms lower bound OPT by finding high-value solutions.

To convert sum-of-squares into a search algorithm, one needs to design a *rounding algorithm* that takes a pseudodistribution and outputs an actual feasible point. This is one of the important, non-black-box aspects of the sum-of-squares algorithm, but we won't study rounding algorithms in this thesis.

A sum-of-squares proof is simply a polynomial identity of a special form.

Definition 3.14 (Sum-of-squares proof)**.** *A degree-D sum-of-squares proof of $p(\mathsf{X}) \geq 0$ from axioms $\{f_i(\mathsf{X}) = 0, g_j(\mathsf{X}) \geq 0\}$ is a polynomial equality*

$$p(\mathsf{X}) = \sum_i f_i(\mathsf{X})p_i(\mathsf{X}) + q_0(\mathsf{X}) + \sum_j g_j(\mathsf{X})q_j(\mathsf{X})$$

where each $q_j(\mathsf{X}) = \sum_k q_{j,k}(\mathsf{X})^2$ is a sum of square polynomials, and all terms have degree at most D.

We write $\mathcal{A} \vdash_D C$ if there is a degree-D proof of inequality C from $\mathcal{A}$, and $\vdash_D C$ if $\mathcal{A} = \emptyset$.

Definition 3.15 (Sum-of-squares refutation)**.** *A degree-D sum-of-squares refutation of a polynomial system $\mathcal{A}$ is a proof $\mathcal{A} \vdash_D -1 \geq 0$.*

The fundamental duality theorem of SoS is that for a polynomial system $\mathcal{A}$ and all $D \in \mathbb{N}$, exactly one of the following holds (under suitable assumptions on $\mathcal{A}$, such as the inclusion of Boolean axioms "$\mathsf{X}_i^2 = 1$"):

(i) there exists a degree-D sum-of-squares refutation of $\mathcal{A}$,

(ii) there exists an SoS-feasible degree-D pseudoexpectation operator that satisfies $\mathcal{A}$.

Furthermore, the semidefinite program in Definition 3.12 can be used to find whichever one exists. A formal statement and proof can be found in [FKP19, Section 3.1.3] or [JH16].

Recall that our overall goal is to analyze the sum-of-squares algorithm, i.e. the existence or non-existence of a pseudoexpectation operator $\widetilde{\mathbb{E}}$. The interaction between SoS proofs and $\widetilde{\mathbb{E}}$ is that valid $\widetilde{\mathbb{E}}$ must respect all SoS-provable equalities and inequalities.

Proposition 3.16. *If $\widetilde{\mathbb{E}}$ is an SoS-feasible degree-D pseudoexpectation that satisfies $\mathcal{A} = \{f_i(\mathsf{X}) = 0, g_j(\mathsf{X}) \geq 0\}$ and $\mathcal{A} \vdash_D p(\mathsf{X}) \geq 0$, then $\widetilde{\mathbb{E}}[p(\mathsf{X})] \geq 0$.*

Proof. The proof of "$p(\mathsf{X}) \geq 0$" looks like:

$$p(\mathsf{X}) = \sum_i f_i(\mathsf{X}) p_i(\mathsf{X}) + \sum_k q_{0,k}(\mathsf{X})^2 + \sum_{j,k} g_j(\mathsf{X}) q_{j,k}(\mathsf{X})^2 .$$

Applying $\widetilde{\mathbb{E}}$ to both sides (which is linear),

$$\begin{aligned}\widetilde{\mathbb{E}}[p(\mathsf{X})] &= \sum_i \underbrace{\widetilde{\mathbb{E}}[f_i(\mathsf{X})p_i(\mathsf{X})]}_{\substack{=0 \\ (\widetilde{\mathbb{E}} \text{ satisfies } ``f_i(\mathsf{X}) = 0")}} + \sum_k \underbrace{\widetilde{\mathbb{E}}[q_{0,k}(\mathsf{X})^2]}_{\substack{\geq 0 \\ \text{(SoS-feasible)}}} + \sum_{j,k} \underbrace{\widetilde{\mathbb{E}}[g_j(\mathsf{X})q_{j,k}(\mathsf{X})^2]}_{\substack{\geq 0 \\ (\widetilde{\mathbb{E}} \text{ satisfies } ``g_j(\mathsf{X}) \geq 0")}} \\ &\geq 0\end{aligned}$$

□

We may use this proposition to conclude many useful facts about $\widetilde{\mathbb{E}}$ using sum-of-squares reasoning (we may also use linearity of expectation, since $\widetilde{\mathbb{E}}$ is linear). If we pretend $\widetilde{\mathbb{E}}$ is a true distribution $\mathbb{E}$ of solutions, and prove an (in)equality about $\mathbb{E}$ using an SoS proof, then $\widetilde{\mathbb{E}}$ must also satisfy the (in)equality.[2] SoS captures many common arguments, such as Cauchy-Schwarz and hypercontractive inequalities [KOTZ14]. For example, $\mathbb{E}[\mathsf{X}^2] \geq \mathbb{E}[\mathsf{X}]^2$ follows from the non-negativity of variance, $\mathbb{E}[(\mathsf{X} - \mathbb{E}[\mathsf{X}])^2] \geq 0$, which is non-negativity of a square polynomial, and therefore the inequality $\widetilde{\mathbb{E}}[\mathsf{X}^2] \geq \widetilde{\mathbb{E}}[\mathsf{X}]^2$ is satisfied by any SoS-feasible pseudoexpectation operator. We give a more extended example in Fact 3.20 after a discussion of dynamic vs. static proof systems.

SoS is a *static* proof system, meaning that a proof is given all at once in one line. This can be contrasted with a *dynamic* proof system, where deduction rules are applied one line after the next, and which reflects more accurately how proofs are written in the real world. We write SoS proofs in the dynamic SoS proof system since it's more natural to follow. The two are *not* equivalent (the dynamic proof system is stronger [AH18]), but usually a proof in the dynamic system can be converted into a static proof in the natural way, by just compressing all of the derivations into a single line.

[2] Formally, in Proposition 3.16 we will frequently use polynomials $p(\mathsf{X})$ whose coefficients may depend on $\widetilde{\mathbb{E}}$ itself. We write "$\mathcal{A} \vdash_D$ (an inequality involving $\mathbb{E}$)" to emphasize that the pseudoexpectation operator emulates this property of a real expectation operator, although formally it should be understood as "any degree-D pseudoexpectation operator that satisfies $\mathcal{A}$ also satisfies (an inequality involving $\widetilde{\mathbb{E}}$)".

The dynamic SoS proof system works by deriving new equalities and inequalities from given ones. However, we are not permitted to apply any type of reasoning (e.g. calculus) to the known inequalities. We may only use non-negativity of squares along with a couple of other simple deductions.

Definition 3.17 (Dynamic sum-of-squares proof system)**.** *In the dynamic degree-D sum-of-squares proof system, we start from an initial set of polynomial axioms (augmented with $0 = 0$) and may derive new equalities and inequalities. Given the set of currently-deduced equalities "$f_i(\mathsf{X}) = 0$" and inequalities "$g_j(\mathsf{X}) \geq 0$", the next line of the proof is permitted to be one of the following deductions:*

(i) $p(\mathsf{X})^2 \geq 0$, *where* $2\deg(p) \leq D$

(ii) $f_i(\mathsf{X})p(\mathsf{X}) = 0$, *where* $\deg(f_i) + \deg(p) \leq D$

(iii) $g_j(\mathsf{X})p(\mathsf{X})^2 \geq 0$, *where* $\deg(g_j) + 2\deg(p) \leq D$

(iv) $f_{i_1}(\mathsf{X}) + f_{i_2}(\mathsf{X}) = 0$

(v) $g_{j_1}(\mathsf{X}) + g_{j_2}(\mathsf{X}) \geq 0$

(vi) $f_i(\mathsf{X}) + g_j(\mathsf{X}) \geq 0$

We have said that the SoS proof system has complete local reasoning, and what this means is that any true polynomial equality or inequality on a small set (at most $D/2$) of variables may be deduced and hence utilized in the proof. Correspondingly, any valid $\widetilde{\mathbb{E}}$ must obey all locally true (in)equalities on at most $D/2$ variables.

Proposition 3.18. *Given is a set of polynomial constraints $\mathcal{A}$ on variables $\mathsf{X}_1, \ldots, \mathsf{X}_n$ such that $\mathcal{A}$ includes "$\mathsf{X}_i^2 = 1$" for all i. For $S \subseteq [n]$, let $[\![\mathcal{A}]\!]_S$ be the subset of axioms $\mathcal{A}$ that only involve variables in S. If C is a polynomial (in)equality that also involves only variables in S, such that C holds for all feasible points X of $[\![\mathcal{A}]\!]_S$, then any SoS-feasible degree-$2|S|$ pseudoexpectation operator $\widetilde{\mathbb{E}}$ that satisfies $\mathcal{A}$ also has $\widetilde{\mathbb{E}}[C] = 0$ (or $\widetilde{\mathbb{E}}[C] \geq 0$).*

Proof. By Boolean Fourier inversion, there exists a local distribution of solutions φ on $\{-1, +1\}^S$ that agrees with $\widetilde{\mathbb{E}}$ on S [FKP19, Section 3.2.2.2]. We have $\widetilde{\mathbb{E}}[C] = \mathbb{E}_{x\sim\varphi}[C(x)]$. The local distribution φ is necessarily supported on feasible points of $[\![\mathcal{A}]\!]_S$, thus the claim holds. □

Example 3.19. *Given a graph E on vertex set $[n]$, the Max Cut problem is:*

$$\boxed{\begin{array}{lll} & \text{Variables: } \mathsf{X}_1, \ldots, \mathsf{X}_n & \\ \max & \displaystyle\mathop{\mathbb{E}}_{\{i,j\}\in_{\mathrm{R}} E}\left[\frac{1}{2} - \frac{1}{2}\mathsf{X}_i\mathsf{X}_j\right] & \\ \text{s.t.} & \mathsf{X}_i^2 = 1 & \forall i = 1, \ldots, n \end{array}}$$

The "triangle constraints" observe that for every ± 1 *Boolean assignment to three variables* $\mathsf{X}_i, \mathsf{X}_j, \mathsf{X}_k$*,*

$$\begin{aligned} \mathsf{X}_i\mathsf{X}_j + \mathsf{X}_i\mathsf{X}_k + \mathsf{X}_j\mathsf{X}_k &\geq -1 \\ \mathsf{X}_i\mathsf{X}_j - \mathsf{X}_i\mathsf{X}_k - \mathsf{X}_j\mathsf{X}_k &\geq -1\,. \end{aligned}$$

These constraints involve three distinct variables, hence degree-6 SoS implies these constraints (in fact, they can be proven using degree-4 SoS). Adding these constraints to the Goemans-Williamson relaxation for Max Cut can improve the performance of the SDP on certain graphs [FS02]. However, O'Donnell and Wu [OW08] proved that adding the triangle constraints does not reduce the integrality gap of the Goemans-Williamson SDP.

We can SoS-prove a polynomial version of the monotonicity of moments. In this proof, X denotes a single variable.

Fact 3.20. *For* $k \in \mathbb{N}$, $\vdash_{2k} \mathbb{E}[\mathsf{X}^{2k}] \geq \mathbb{E}[\mathsf{X}^2]^k$ *and* $\mathsf{X} \geq 0 \vdash_k \mathbb{E}[\mathsf{X}^k] \geq \mathbb{E}[\mathsf{X}]^k$*.*

Proof. The first inequality actually follows from the second. SoS proofs are highly compositional. Given a degree-D proof of the second inequality, if we plug in a degree-d polynomial $p(\mathsf{X})$ for X, we produce a degree-dD proof of the corresponding claim for $p(\mathsf{X})$. Doing this for $p(\mathsf{X}) = \mathsf{X}^2$, since $\vdash_2 p(\mathsf{X}) \geq 0$, we may remove the assumption "$p(\mathsf{X}) \geq 0$".

The second inequality follows from a slightly more general inequality of moments (itself a consequence of Hölder's inequality).

Fact 3.21. *For* $k, \ell \in \mathbb{N}$*,*

$$\mathsf{X} \geq 0 \vdash_{k+\ell} \mathbb{E}\left[\mathsf{X}^{k+\ell}\right] \geq \mathbb{E}\left[\mathsf{X}^k\right] \cdot \mathbb{E}\left[\mathsf{X}^\ell\right].$$

Proof. We induct on $k + \ell$. WLOG, $k \geq \ell > 0$.

In the first case, $k + \ell$ is even. Consider the following square polynomial,

$$\begin{aligned} S(\mathsf{X}) &= \mathsf{X}^{\frac{k+\ell}{2}} - \mathsf{X}^{\frac{k-\ell}{2}}\,\mathbb{E}[\mathsf{X}^\ell]\,, \\ \mathbb{E}[S(\mathsf{X})^2] &= \mathbb{E}[(\mathsf{X}^{\frac{k+\ell}{2}} - \mathsf{X}^{\frac{k-\ell}{2}}\,\mathbb{E}[\mathsf{X}^\ell])^2] \\ &= \mathbb{E}[\mathsf{X}^{k+\ell}] - 2\,\mathbb{E}[\mathsf{X}^k]\,\mathbb{E}[\mathsf{X}^\ell] + \mathbb{E}[\mathsf{X}^{k-\ell}]\,\mathbb{E}[\mathsf{X}^\ell]^2 \\ &= \mathbb{E}[\mathsf{X}^{k+\ell}] - \Big(2\,\mathbb{E}[\mathsf{X}^k] - \mathbb{E}[\mathsf{X}^{k-\ell}]\,\mathbb{E}[\mathsf{X}^\ell]\Big)\,\mathbb{E}[\mathsf{X}^\ell]\,. \end{aligned}$$

In the second case, $k + \ell$ is odd. Letting $S(\mathsf{X})$ be the same polynomial for $k + \ell - 1$, we recover the same expression.

$$\mathbb{E}[\mathsf{X}S(\mathsf{X})^2] = \mathbb{E}[\mathsf{X}^{k+\ell}] - \Big(2\,\mathbb{E}[\mathsf{X}^k] - \mathbb{E}[\mathsf{X}^{k-\ell}]\,\mathbb{E}[\mathsf{X}^\ell]\Big)\,\mathbb{E}[\mathsf{X}^\ell]\,.$$

By the induction hypothesis, there is an SoS proof that the term inside the parentheses is at least $\mathbb{E}[\mathsf{X}^k]$. Plugging in this proof and multiplying by the (provably non-negative) number $\mathbb{E}[\mathsf{X}^\ell]$,

$$\mathsf{X} \geq 0 \vdash_k \left(2\,\mathbb{E}[\mathsf{X}^k] - \mathbb{E}[\mathsf{X}^{k-\ell}]\,\mathbb{E}[\mathsf{X}^\ell]\right)\mathbb{E}[\mathsf{X}^\ell] \geq \mathbb{E}[\mathsf{X}^k]\cdot\mathbb{E}[\mathsf{X}^\ell]\,.$$

$$\mathsf{X} \geq 0 \vdash_{k+\ell} \mathbb{E}[S(\mathsf{X})^2] \geq 0 \qquad\qquad \mathsf{X} \geq 0 \vdash_{k+\ell} \mathbb{E}[\mathsf{X}S(\mathsf{X})^2] \geq 0\,.$$

Adding these two inequalities, we have the claim. □

To use this fact to prove Fact 3.20, apply induction to remove one power of $\mathbb{E}[\mathsf{X}]$ at a time. □

3.2 Pseudocalibration

A typical use case of sum-of-squares is for statistical distinguishing problems. In this setting we are given a sample from either a *random distribution* or a *planted distribution* (also known as a *null distribution* and *alternative distribution*) and we are attempting to determine which case the sample was drawn from.

SoS can be used as a distinguishing algorithm by setting up a polynomial system that encodes the planted structure. SoS distinguishes the two distributions if it can refute feasibility of the planted structure when given a random instance. What is surprising is that, for statistical distinguishing problems, there seem to be canonically optimal choices for some of the inherent choices of the SoS algorithm. What polynomial system should be used? Banks et al [BMR21] suggest that the *local statistics hierarchy* is a canonical choice.

When proving a lower bound, a suggested canonical method for constructing a candidate pseudoexpectation operator is *pseudocalibration*. Recall that the goal of a lower bound is to construct a fake distribution $\widetilde{\mathbb{E}}$ of planted structures when given a sample G from the random distribution (which usually does not contain the planted structure whp). The main idea of pseudocalibration is that, we should "pretend" that G comes from the planted case, and let $\widetilde{\mathbb{E}}$ be the marginal distribution of the planted structure given the instance G.

Of course, since there is no planted structure in G whp, the marginal distribution is ill-defined. To get around this, we syntactically restrict $\widetilde{\mathbb{E}}$ to be only the "low-degree" part of the marginal distribution. Intuitively, as long as this "low-degree" is significantly larger than the SoS degree (usually by a multiplicative constant or $\log n$ factor), then the SoS algorithm should not notice that we are using a truncation of the marginal distribution.[3]

Now we give the formal definition.

[3] We will need to prove that the density function of the marginal distribution behaves nicely under truncation. For example, the total level-i Fourier weight should decay in i (but this can only be shown for small i, since for large i the truncation approaches the true marginal distribution, which is extremely spiky). Good behavior of the truncation seems to be the case in many applications, aligning with the intuition that "higher-degree does not help the distinguisher". See [Hop18, end of Section 3.3 and Chapter 4] for some further discussion.

Definition 3.22 ($\mathcal{D}_{\mathrm{ra}}$ and $\mathcal{D}_{\mathrm{pl}}$)**.** *"$G \sim \mathcal{D}_{\mathrm{ra}}$" and "$(x, G) \sim \mathcal{D}_{\mathrm{pl}}$" denote the random and planted distributions. In the planted distribution, the input is G and x denotes the planted structure in that instance. For example, in Planted Clique, $x \in \{0,1\}^{V(G)}$ would be the indicator vector of the planted clique.*

We will use the shorthand $\mathbb{E}_{\mathrm{pl}}, \mathbb{E}_{\mathrm{ra}}, \Pr_{\mathrm{pl}}, \Pr_{\mathrm{ra}}$.

In the planted distribution, the marginal distribution of x given G is, by Bayes' theorem:

$$\Pr_{\mathrm{pl}}[x \mid G] = \frac{\Pr_{\mathrm{pl}}[x, G]}{\Pr_{\mathrm{pl}}[G]} = \frac{\Pr_{\mathrm{pl}}[x, G]}{\sum_x \Pr_{\mathrm{pl}}[x, G]},$$

$$\mathop{\mathbb{E}}_{\mathrm{pl}}[p(x) \mid G] = \frac{\sum_x \Pr_{\mathrm{pl}}[x, G] p(x)}{\sum_x \Pr_{\mathrm{pl}}[x, G]}.$$

Observe that if G does not have any occurrences of the planted structure, then all terms are 0 and the distribution is ill-defined. Pseudocalibration suggests truncating each of the expressions $\Pr_{\mathrm{pl}}[x, G]$ to be a low-degree polynomial in G. Actually, so that the numerator and denominator are normalized better, we truncate the *likelihood ratio* $\Pr_{\mathrm{pl}}[x, G] / \Pr_{\mathrm{ra}}[G]$ (a.k.a. the *Radon-Nikodym derivative*), noting the identity:

$$\mathop{\mathbb{E}}_{\mathrm{pl}}[p(x) \mid G] = \frac{\sum_x \Pr_{\mathrm{pl}}[x, G] p(x) / \Pr_{\mathrm{ra}}[G]}{\sum_x \Pr_{\mathrm{pl}}[x, G] / \Pr_{\mathrm{ra}}[G]}.$$

Formally the truncation is accomplished by viewing G as a point in $\mathbb{R}^N$ and, with x fixed, truncating the function $\frac{\Pr_{\mathrm{pl}}[x,G]}{\Pr_{\mathrm{ra}}[G]} : \mathbb{R}^N \to \mathbb{R}$ to have degree at most τ. Concretely, let χ_α be the Fourier basis for G. Expanding the numerator in the Fourier basis,

$$\begin{aligned}
&\sum_x p(x) \frac{\Pr_{\mathrm{pl}}[x, G]}{\Pr_{\mathrm{ra}}[G]} \\
&= \sum_x p(x) \sum_\alpha \widehat{\frac{\Pr_{\mathrm{pl}}[x, G]}{\Pr_{\mathrm{ra}}[G]}}(\alpha) \cdot \frac{\chi_\alpha(G)}{\mathop{\mathbb{E}}_{G' \sim \mathrm{ra}}[\chi_\alpha(G')^2]} && \text{(Fourier inversion, Fact 1.7)} \\
&= \sum_x p(x) \sum_\alpha \mathop{\mathbb{E}}_{G' \sim \mathrm{ra}} \left[\frac{\Pr_{\mathrm{pl}}[x, G']}{\Pr_{\mathrm{ra}}[G']} \chi_\alpha(G') \right] \cdot \frac{\chi_\alpha(G)}{\mathop{\mathbb{E}}_{G' \sim \mathrm{ra}}[\chi_\alpha(G')^2]} \\
&= \sum_x p(x) \sum_\alpha \sum_{G'} \Pr_{\mathrm{pl}}[x, G'] \chi_\alpha(G') \cdot \frac{\chi_\alpha(G)}{\mathop{\mathbb{E}}_{G' \sim \mathrm{ra}}[\chi_\alpha(G')^2]} \\
&= \sum_\alpha \mathop{\mathbb{E}}_{(x, G') \sim \mathrm{pl}}[p(x) \chi_\alpha(G')] \cdot \frac{\chi_\alpha(G)}{\mathop{\mathbb{E}}_{G' \sim \mathrm{ra}}[\chi_\alpha(G')^2]}.
\end{aligned}$$

Therefore we are led to the following definition. Let $|\alpha|$ be the degree of the Fourier character χ_α.

Definition 3.23 (Pseudocalibration). *Given an input $G \in \mathbb{R}^N$ and a truncation parameter $\tau \in \mathbb{N}$, the pseudocalibrated pseudoexpectation operator $\widetilde{\mathbb{E}}$ is:*

$$\widetilde{\mathbb{E}}[p(\mathsf{X})] = \sum_{\alpha : |\alpha| \leq \tau} \mathop{\mathbb{E}}_{(x, G') \sim \mathcal{D}_{\text{pl}}} [p(x)\chi_\alpha(G')] \cdot \frac{\chi_\alpha(G)}{\mathbb{E}_{G' \sim \mathcal{D}_{\text{ra}}}[\chi_\alpha(G')^2]}$$

and then divide the operator by $\widetilde{\mathbb{E}}[1]$.

The Fourier coefficients $\mathbb{E}_{(x,G')\sim\mathcal{D}_{\text{pl}}}[p(x)\chi_\alpha(G')]$ can usually be explicitly computed, which therefore gives an explicit pseudoexpectation operator $\widetilde{\mathbb{E}}$.

An advantage of pseudocalibration is that this construction automatically satisfies some nice properties that the pseudoexpectation $\widetilde{\mathbb{E}}$ should satisfy. $\widetilde{\mathbb{E}}[p]$ is linear in p by construction. For all polynomial equalities of the form "$f(\mathsf{X}) = 0$" that are true with probability 1 in the planted distribution, we will have that $\widetilde{\mathbb{E}}[f(\mathsf{X})] = 0$. For other polynomial equalities of the form "$f(\mathsf{X}, G) = 0$" that are satisfied in the planted distribution, the equality $\widetilde{\mathbb{E}}[f(\mathsf{X}, G)] = 0$ is approximately satisfied, up to truncation error τ. Furthermore, $\widetilde{\mathbb{E}}$ can be tweaked to exactly satisfy these constraints. This is described in the next subsection.

Fact 3.24 (Proof in [GJJ$^+$20, Section 7]). *If $p(\mathsf{X})$ is a polynomial which is uniformly zero on the planted distribution, then $\widetilde{\mathbb{E}}[p(\mathsf{X})]$ is the zero function. If $p(\mathsf{X}, G)$ is a polynomial which is uniformly zero on the planted distribution, then the only nonzero Fourier coefficients of $\widetilde{\mathbb{E}}[p(\mathsf{X}, G)]$ are those with degree in the range $\tau \pm \deg_G(p)$.*

As for inequality constraints "$g(\mathsf{X}) \geq 0$" that are satisfied in the planted distribution, showing that $\widetilde{\mathbb{E}}$ satisfies these is the main technical PSD-ness argument of the proofs.

When pseudocalibrating, although the truncation to low-degree is uniquely defined, one can freely choose the "inner" basis of low-degree functions χ_α. For example, in settings with additional symmetry, it may make sense to use a symmetrized basis. For the problem solved in Chapter 4, we attempted to use the alternate basis described in Chapter 5 to simplify the analysis, though ultimately we reverted to the standard Fourier basis.

We remark that the normalization factor $\widetilde{\mathbb{E}}[1]$ in Definition 3.23 is the *low-degree likelihood ratio*. In all successful applications of pseudocalibration, $\widetilde{\mathbb{E}}[1] = 1 \pm o(1)$, and hence dividing by $\widetilde{\mathbb{E}}[1]$ does not significantly change the value of the pseudoexpectation operator. In fact, whether or not $\widetilde{\mathbb{E}}[1] = 1 \pm o(1)$ has been quite successful in predicting polynomial time distinguishability of the planted and random distributions. See the open problems at the end of the chapter.

3.2.1 Satisfying equality constraints exactly

If we have a pseudoexpectation operator that almost satisfies a polynomial constraint "$f(\mathsf{X}) = 0$", then we can round the operator slightly so that it exactly satisfies the constraint. This is formally accomplished by viewing $\widetilde{\mathbb{E}} \in \mathbb{R}^{\left(\!\binom{[n]}{\le D}\!\right)}$ as a vector and expressing the constraints as a matrix Q such that $\widetilde{\mathbb{E}}$ satisfies the constraints if and only if it lies in the null space of Q. To round the operator, we project $\widetilde{\mathbb{E}}$ to $\mathrm{Null}(Q)$.

Letting c_α be the coefficient of f on X^α, the choice of the matrix $Q \in \mathbb{R}^{\left(\!\binom{[n]}{\le D-\deg(f)}\!\right) \times \left(\!\binom{[n]}{\le D}\!\right)}$ is:

$$Q[I,J] = \begin{cases} c_{J\setminus I} & J \supseteq I \\ 0 & \text{otherwise} \end{cases}.$$

To satisfy multiple constraints, we may stack the matrices Q. Projection to $\mathrm{Null}(Q)$ is accomplished by the explicit projection matrix

$$I - Q^\intercal (QQ^\intercal)^+ Q$$

where $(QQ^\intercal)^+$ is the pseudo-inverse of matrix $QQ^\intercal$, which is not invertible. To prove that the rounding is not significant, we must bound the norm of the second term. This amounts to proving a lower bound on the minimum nonzero eigenvalue of the matrix $QQ^\intercal$.

Remark 3.25. *When the constraint depends on G, the projection matrix Q is a function of G, in which case Q can be analyzed in terms of graph matrices. Recall that pseudocalibration guarantees that constraints that don't depend on G are already exactly satisfied.*

As an example, we can project $\widetilde{\mathbb{E}}$ so that it satisfies the Boolean constraints "$\mathsf{X}_i^2 = 1$" (though normally this is enforced definitionally by Remark 3.3, or by pseudocalibration). Let Q_{bool} be the "check matrix" for the Boolean constraints "$\mathsf{X}_i^2 = 1$". Q_{bool} has $n \cdot \left(\!\binom{n}{\le D-2}\!\right)$ rows. The (i,α) row checks $\widetilde{\mathbb{E}}[\mathsf{X}^\alpha \cdot \mathsf{X}_i^2] = \widetilde{\mathbb{E}}[\mathsf{X}^\alpha]$. It has entry 1 in column α and entry -1 in column $\alpha \cup \{i,i\}$.

Lemma 3.26. *Assume that $\widetilde{\mathbb{E}}$ approximately satisfies the boolean constraints:*

$$\widetilde{\mathbb{E}}[\mathsf{X}^\alpha \cdot (\mathsf{X}_i^2 - 1)] \le \varepsilon$$

for any X^α with degree at most $D-2$. Then letting $\widetilde{\mathbb{E}}'$ be the projection to $\mathrm{Null}(Q_{bool})$*, we have*

$$\left\| \widetilde{\mathbb{E}} - \widetilde{\mathbb{E}}' \right\|_2 \le \varepsilon n^{O(D)}.$$

Proof. The effect of projecting $\widetilde{\mathbb{E}}$ to $\mathrm{Null}(Q_{bool})$ is to symmetrize $\widetilde{\mathbb{E}}[\mathsf{X}^{\alpha+2\beta}]$ across all β; average all entries $\widetilde{\mathbb{E}}[1], \widetilde{\mathbb{E}}[\mathsf{X}_1^2], \widetilde{\mathbb{E}}[\mathsf{X}_2^2], \widetilde{\mathbb{E}}[\mathsf{X}_1^6\mathsf{X}_7^4\mathsf{X}_{10}^2]$ etc, average $\widetilde{\mathbb{E}}[\mathsf{X}_1], \widetilde{\mathbb{E}}[\mathsf{X}_1\mathsf{X}_3^2], \widetilde{\mathbb{E}}[\mathsf{X}_1\mathsf{X}_3^4\mathsf{X}_4^4]$ etc, and so on. One can see this because this is a linear map which fixes $\mathrm{Null}(Q_{bool})$ and takes all vectors into $\mathrm{Null}(Q_{bool})$.

By assumption, there is additive error ε between $\widetilde{\mathbb{E}}[\mathsf{X}^\alpha]$ and $\widetilde{\mathbb{E}}[\mathsf{X}^\alpha \cdot \mathsf{X}_i^2]$. As the size of β is at most D, we have $\widetilde{\mathbb{E}}[\mathsf{X}^{\alpha+2\beta}] = \widetilde{\mathbb{E}}[\mathsf{X}^\alpha] \pm \varepsilon D$ for all β. Therefore averaging these entries changes each of them by at most εD. Overall,

$$\begin{aligned}\left\|\widetilde{\mathbb{E}} - \widetilde{\mathbb{E}}'\right\|_2 &\leq \left(\!\!\binom{n}{\leq D}\!\!\right) \cdot \left\|\widetilde{\mathbb{E}} - \widetilde{\mathbb{E}}'\right\|_\infty \\ &\leq n^{O(D)} \cdot \varepsilon D = \varepsilon n^{O(D)}.\end{aligned}$$

□

To bound the effect of projection on the moment matrix, we have the following proposition.

Proposition 3.27. *Given two pseudoexpectation operators* $\widetilde{\mathbb{E}}, \widetilde{\mathbb{E}}' \in \mathbb{R}^{\left(\!\binom{[n]}{\leq D}\!\right)}$,

$$\left\|\mathcal{M}(\widetilde{\mathbb{E}}) - \mathcal{M}(\widetilde{\mathbb{E}}')\right\| \leq n^{O(D)} \left\|\widetilde{\mathbb{E}} - \widetilde{\mathbb{E}}'\right\|_\infty \leq n^{O(D)} \left\|\widetilde{\mathbb{E}} - \widetilde{\mathbb{E}}'\right\|_2.$$

Proof. The norm of $\mathcal{M}(\widetilde{\mathbb{E}}) - \mathcal{M}(\widetilde{\mathbb{E}}')$ is bounded by the maximum row sum. The entries of $\mathcal{M}(\widetilde{\mathbb{E}}) - \mathcal{M}(\widetilde{\mathbb{E}}')$ are bounded in magnitude by $\left\|\widetilde{\mathbb{E}} - \widetilde{\mathbb{E}}'\right\|_\infty$ while the dimension is $n^{O(D)}$, so the claim follows. □

We can use this rounding technique with a pseudocalibrated $\widetilde{\mathbb{E}}$. The low-degree truncation used in the definition of pseudocalibration, Definition 3.23, introduces a tiny error in constraints which depend on the instance G. By taking the truncation parameter τ to be larger than the degree D of the SoS solution, the truncation error is small enough so that the rounding barely affects the spectrum of the moment matrix. See [GJJ+20, Lemma 7.7] for a quantitative bound.

3.3 Proving PSD-ness using graph matrices

Pseudocalibration suggests a candidate pseudoexpectation operator/moment matrix that satisfies (or almost satisfies) the problem constraints. What remains is to prove that the moment matrix is PSD. The pseudocalibration formula in Definition 3.23 gives us the Fourier coefficients of $\widetilde{\mathbb{E}}$, which expresses the moment matrix in terms of ribbons from Chapter 2. The ribbons can be grouped into graph matrices if there is symmetry in the Fourier coefficients of $\widetilde{\mathbb{E}}$, which is inherited from symmetry of the planted distribution. Here we explain the *approximate PSD decomposition*, a general technique to prove PSD-ness of the moment matrix once it is expressed in the graph matrix basis.

It is possible that pseudocalibration plus a suitably general application of this technique could be a universal lower bound technique for SoS on statistical distinguishing problems. This is surprising, and partial progress is made in the tour-de-force work [PR20], which gives sufficient conditions for an SoS lower bound. However, we do not state a concrete conjecture about universality of the technique. (For various technical reasons, we are not able to formally apply [PR20] in later chapters.)

We will not be too concerned with the specific definition of graph matrices in this section, expressing the moment moment in an abstract way for some coefficients λ_α,

$$\Lambda = \sum_{\text{shapes } \alpha} \lambda_\alpha M_\alpha \,.$$

A first attempt to show $\Lambda \succeq 0$, which works sometimes, is to bound all terms against $\mathbb{E}\,\Lambda$.

$$\mathbb{E}\,\Lambda = \sum_{\substack{\text{shapes } \alpha: \\ E(\alpha)=\emptyset}} \lambda_\alpha M_\alpha \,.$$

It must be checked that $\mathbb{E}\,\Lambda \succeq 0$. To show that the spectrum of $\mathbb{E}\,\Lambda$ dominates the spectrum of Λ, we can use the triangle inequality then the graph matrix norm bounds to prove that:

$$\|\Lambda - \mathbb{E}\,\Lambda\| \leq \sum_{\substack{\text{shapes } \alpha: \\ E(\alpha)\neq\emptyset}} |\lambda_\alpha| \|M_\alpha\| \leq \mathrm{o}(\text{minimum eigenvalue of } \mathbb{E}\,\Lambda) \,.$$

In the presence of a null space for $\mathbb{E}\,\Lambda$, the situation is more delicate. We are able to use this simpler approach and handle the null space in Chapter 4.

The approximate PSD decomposition is a more sophisticated approach that leverages the combinatorial structure of graph matrices. We will use this approach in Chapter 6. Recall that our goal is to show that the matrix $\sum_\alpha \lambda_\alpha M_\alpha$ is PSD. Each shape α decomposes into left, middle, and right parts $\alpha = \sigma \circ \tau \circ \sigma'^{\mathsf{T}}$. We have a corresponding approximate decomposition of the graph matrix for α, up to intersection terms,

$$M_\alpha = M_{\sigma \circ \tau \circ \sigma'^{\mathsf{T}}} \approx M_\sigma M_\tau M_{\sigma'}^{\mathsf{T}} \,.$$

The dominant term in the PSD decomposition of Λ collects together α such that its middle shape τ is trivial. We assume for simplicity that the coefficients λ_α factor, $\lambda_{\alpha\circ\beta} = \lambda_\alpha \cdot \lambda_\beta$. The dominant term is approximately PSD:

$$\sum_{\alpha: \text{ trivial}} \lambda_\alpha M_\alpha = \sum_{\sigma,\sigma'} \lambda_{\sigma\circ\sigma'^{\mathsf{T}}} M_{\sigma\circ\sigma'^{\mathsf{T}}} \approx \sum_{\sigma,\sigma'} \lambda_\sigma \lambda_{\sigma'} M_\sigma M_{\sigma'}^{\mathsf{T}} = \left(\sum_\sigma \lambda_\sigma M_\sigma\right)\left(\sum_\sigma \lambda_\sigma M_\sigma\right)^{\mathsf{T}} .$$

There are two types of things to handle: α with nontrivial τ, and intersection terms. For α with a fixed middle shape τ, these should all be charged directly to the dominant term (include τ^{T} so the matrix is symmetric):

$$\left(\sum_\sigma \lambda_\sigma M_\sigma\right)\lambda_\tau(M_\tau+M_\tau^{\mathsf{T}})\left(\sum_\sigma\lambda_\sigma M_\sigma\right)^{\mathsf{T}} \preceq \frac{1}{c(\tau)}\left(\sum_\sigma \lambda_\sigma M_\sigma\right)\left(\sum_\sigma \lambda_\sigma M_\sigma\right)^{\mathsf{T}}$$

where we leave some space $c(\tau)$. Intuitively this is possible because nontrivial τ have smaller coefficients λ_τ as the degree of the Fourier polynomial increases, and the norm of M_τ is controlled due to the factorization of α into left, middle, and right parts. This check amounts to $\sum_{\tau \text{ nontrivial}} \lambda_\tau M_\tau \preceq o(1)\mathrm{Id}$, or essentially equivalently, $\sum_{\tau \text{ nontrivial}}|\lambda_\tau|\|M_\tau\| \le o(1)$.

The intersection terms need to be handled in a recursive way so that their norms can be kept under control: if $\sigma, \tau, \sigma'^{\mathsf{T}}$ intersect to create a shape ζ, then we need to factor out the non-intersecting parts of σ and σ' from the intersecting parts γ, γ'. Informally writing

$$\zeta = (\sigma - \gamma) \circ \tau_P \circ (\sigma' - \gamma')^{\mathsf{T}},$$

where τ_P is the intersection of $\gamma, \tau, \gamma'^{\mathsf{T}}$, we now perform a further factorization

$$M_\zeta \approx M_{\sigma-\gamma} M_{\tau_P} M_{\sigma'-\gamma'}^{\mathsf{T}}$$

which recursively creates more intersection terms. The point is that the sum over $\sigma - \gamma, \sigma' - \gamma'$ is equivalent to the sum over σ, σ' (up to truncation error).

$$\begin{aligned}\sum_{\sigma-\gamma}\sum_{\sigma'-\gamma'} \lambda_{\sigma\circ\tau\circ\sigma'^{\mathsf{T}}} M_{\sigma-\gamma} M_{\tau_P} M_{\sigma'-\gamma'}^{\mathsf{T}} &= \left(\sum_{\sigma-\gamma} \lambda_{\sigma-\gamma} M_{\sigma-\gamma}\right) \lambda_{\gamma\circ\tau\circ\gamma'^{\mathsf{T}}} M_{\tau_P} \left(\sum_{\sigma-\gamma} \lambda_{\sigma-\gamma} M_{\sigma-\gamma}^{\mathsf{T}}\right) \\ &= \left(\sum_\sigma \lambda_\sigma M_\sigma\right) \lambda_{\gamma\circ\tau\circ\gamma'^{\mathsf{T}}} M_{\tau_P} \left(\sum_\sigma \lambda_\sigma M_\sigma^{\mathsf{T}}\right) + \text{truncation error.}\end{aligned}$$

In summary, we have the following informal decomposition of the moment matrix,

$$\Lambda = \left(\sum_\sigma \lambda_\sigma M_\sigma\right)\left(\mathrm{Id} + \sum_{\tau \text{ nontrivial}} \lambda_\tau M_\tau + \sum_{\substack{\text{intersection terms} \\ \tau_P \in \mathcal{P}_{\gamma,\tau,\gamma'}}} \lambda_{\gamma\circ\tau\circ\gamma'^{\mathsf{T}}} M_{\tau_P}\right)\left(\sum_\sigma \lambda_\sigma M_\sigma\right)^{\mathsf{T}} + \text{truncation error}\,.$$

We then need to compare the intersection terms M_{τ_P} with Id. The shape τ_P is formed from a middle shape with additional constrained intersections. The norm should be bounded similarly to a middle shape, using an "intersection tradeoff lemma" to show that intersections do not increase the norm too much. We also need to combinatorially bound the number of ways to produce τ_P as an intersection pattern.

Finally, there is the issue of truncation. The total norm of the truncation error should be small. This can be accomplished by taking a large truncation parameter relative to the SoS degree.

If the approximate PSD decomposition described above succeeds, it proves a positive lower bound on the minimum eigenvalue of Λ. If the matrix

$$\mathrm{Id} + \sum_{\tau \text{ nontrivial}} \lambda_\tau M_\tau + \sum_{\substack{\text{intersection terms} \\ \tau_P \in \mathcal{P}_{\gamma,\tau,\gamma'}}} \lambda_{\gamma \circ \tau \circ \gamma'^{\intercal}} M_{\tau_P}$$

instead has a nullspace, it must be handled via an additional argument.

3.4 Open Problems

For statistical distinguishing problems, the *low-degree hypothesis* conjectures that for a certain class of problems, polynomial-time distinguishability is achievable if and only if the low-degree likelihood ratio is a successful distinguisher. The conjecture was formulated by Hopkins [Hop18, Conjecture 2.2.4]; for a formal statement and further discussion see [HW20, ZSWB21].

In the context of sum-of-squares, the low-degree hypothesis implies that sum-of-squares is no stronger than low-degree polynomials. A reduction from sum-of-squares to low-degree polynomials has been proven for some settings by Hopkins et al [HKP$^+$17]. However, their proof is not strong enough to formally imply SoS lower bounds in the settings we consider in later chapters, and it would be useful to make their result more composable.

We have mentioned that sum-of-squares is a semidefinite programming meta-algorithm. For Constraint Satisfaction Problems (CSPs), it is known that degree-$O(1)$ sum-of-squares achieves the best approximation ratio of any polynomial-size semidefinite program [LRS15]. However, it seems possible that in general, sum-of-squares may *not* be the optimal choice of semidefinite programming algorithm.

Question 3.28. *Can we give a concrete problem where augmenting the sum-of-squares SDP with additional linear constraints significantly improves performance of the algorithm?*

CSPs are highly local, so it makes sense that SoS_d (which has complete local reasoning) could be an optimal semidefinite program. However, in general, it seems that adding more global constraints could be helpful. Kothari, O'Donnell, and Schramm [KOS19] prove a negative result along these lines for CSPs augmented with a global cardinality constraint.

This open question points out a fundamental deficiency in existing SoS lower bounds, which is that the lower bound doesn't formally apply if the polynomial system is tweaked slightly, for example by binary searching over one of the parameters.

CHAPTER 4
LOWER BOUND FOR THE SHERRINGTON-KIRKPATRICK MODEL

In this chapter we will study the performance of the sum-of-squares algorithm on the generic quadratic optimization problem,

$$\mathsf{OPT}(W) := \max_{x \in \{\pm 1\}^n} x^\mathsf{T} W x, \tag{4.1}$$

where W is a symmetric matrix in $\mathbb{R}^{n \times n}$. In the average-case setting where W has iid standard Gaussian entries, this is known to statistical physicists as the *Sherrington-Kirkpatrick (SK) model*. Some of the most important ideas in statistical physics trace their roots to the SK model. These ideas began with Parisi [Par79], who gave a heuristic argument (proven rigorously by Talagrand [Tal06]) that

$$\mathsf{OPT}(W) \approx 2 \cdot P^* \cdot n^{3/2}$$

where $P^* \approx 0.7632$ is now referred to as the Parisi constant.

We will prove that sum-of-squares cannot certify the value of $\mathsf{OPT}(W)$; specifically, sum-of-squares does no better than the basic spectral algorithm, which can only prove that,

$$\mathsf{OPT}(W) \leq 2 \cdot n^{3/2}.$$

Our technique reduces the problem to what we call the *Planted Affine Planes* problem. This is a hypothesis testing problem in which one is given a collection of vectors $d_1, \dots, d_m \in \mathbb{R}^n$ and the task is to determine whether: (1) the vectors were sampled independently from $\mathcal{N}(0, \mathrm{Id})$, or (2) the vectors are "planted" in two parallel affine planes, i.e. they are sampled from $\mathcal{N}(0, \mathrm{Id})$ conditioned on $\langle v, d_i \rangle^2 = 1$ where $v \sim \mathcal{N}(0, \mathrm{Id})$ is a secret vector. We prove that if the number of vectors m is at most $m \leq n^{1.5-\varepsilon}$ for any $\varepsilon > 0$, then polynomial time sum-of-squares cannot successfully solve this problem.

We were interested in the Planted Affine Planes problem as a stepping stone to studying random Constraint Satisfaction Problems (CSPs). The Planted Affine Planes problem is a "dense problem" that we found as a tractable intermediate step on the way to studying CSPs, which are usually "sparse", i.e. the "constraint graph" has degree $o(n)$ (usually $O(1)$). Amusingly, the SK model was created by physicists for the same reason: it is a "mean-field model" that is physically unrealizable due to being "dense", but mathematically more tractable than the "sparse", realizable models. Generalizing the techniques from dense to sparse settings is a major challenge in both physics and computer science literature. In Chapter 6 we will tackle sum-of-squares on sparse problems.

In this chapter, precise results on the SK model and Planted Affine Planes are stated in Section 4.1. The main sum-of-squares lower bound uses some of the techniques from Chapter 3, as well as some additional techniques, which are sketched in Section 4.2. We pseudocalibrate, Section 4.3, and then carry out the details of the PSD-ness proof, Section 4.4. Section 4.5 reduces the Sherrington-Kirkpatrick lower bound to the Planted Affine Planes lower bound.

Bibliography. Most of the material in this chapter comes from [GJJ+20]. The proof outline, Section 4.2, has been updated to fit the thesis exposition. The key charging arguments have been isolated and significantly cleaned up in Section 4.2.3. Some technical proofs from the paper are omitted.

4.1 Statement of Lower Bounds

Given $W \in \mathbb{R}^{n \times n}$, the general form of a quadratic optimization problem is:

$$\boxed{\begin{array}{rll} & \text{Variables: } \mathsf{X}_1, \ldots, \mathsf{X}_n & \\ \max & \mathsf{X}^\mathsf{T} W \mathsf{X} & \\ \text{s.t.} & \mathsf{X}_i^2 = 1 & \forall i = 1, \ldots, n \end{array}}$$

By taking W to be a graph Laplacian [HLW06, Section 4] the problem is equivalent to the Max Cut problem, a well-known NP-hard problem in the worst case [Kar72]. The worst-case approximation factor of this problem is between $\log^\gamma n$ and $\log n$ for some $\gamma > 0$ [ABE+05, CW04].

Instead of considering $\mathsf{OPT}(W)$ for a worst-case W, one can consider the average-case problem in which W is sampled according to some distribution. One of the simplest models of random matrices is the Gaussian Orthogonal Ensemble, denoted $\mathrm{GOE}(n)$ for n-by-n matrices and defined as follows.

Definition 4.1. *The Gaussian Orthogonal Ensemble, denoted* $\mathrm{GOE}(n)$*, is the distribution of* $\frac{1}{\sqrt{2}}(A + A^\mathsf{T})$ *where A is a random $n \times n$ matrix with i.i.d. standard Gaussian entries.*

In the special case where W is a random matrix drawn from the Gaussian Orthogonal Ensemble, we refer to this as the *Sherrington-Kirkpatrick problem*. We have the following theorem which is the main result of this chapter.

Theorem 4.2. *There exists a constant $\delta > 0$ such that, w.h.p. for $W \sim \mathrm{GOE}(n)$, there is a degree-n^δ SoS solution for the Sherrington–Kirkpatrick problem with value at least* $(2 - o(1)) \cdot n^{3/2}$.

In order to prove Theorem 4.2, we first introduce a new average-case problem we call Planted Affine Planes (PAP) for which we directly prove a SoS lower bound. We then use the PAP lower bound to prove a lower bound on the Sherrington–Kirkpatrick problem.

Definition 4.3 (Planted Affine Planes (PAP) problem). *Given $d_1, \ldots, d_m \in \mathbb{R}^n$, determine whether there exists $v \in \mathbb{R}^n$ such that*

$$\langle v, d_u \rangle^2 = 1,$$

for every $u \in [m]$.

In other words, can we prove that m vectors are not all contained in two parallel hyperplanes at equal distance from the origin? We show that for random vectors $d_u \sim \mathcal{D}$, SoS cannot refute the existence of the parallel hyperplanes. Our results hold for the "Gaussian setting" $\mathcal{D} = \mathcal{N}(0, \mathrm{Id})$ and the "Boolean setting" where $\mathcal{D}$ is uniform over $\{\pm 1\}^n$, though we conjecture (Section 4.6) that similar SoS bounds hold under more general conditions on $\mathcal{D}$.

We will furthermore restrict the solution vector v to be Boolean (scaled appropriately, so that the entries are either $\frac{1}{\sqrt{n}}$ or $\frac{-1}{\sqrt{n}}$). The Boolean restriction on v actually makes the lower bound result stronger since SoS cannot refute even a smaller subset of vectors in $\mathbb{R}^n$.

With the additional Boolean restriction, Planted Affine Planes can be encoded as the feasibility of the polynomial system (given $d_1, \dots, d_m \in \mathbb{R}^n$):

$$\boxed{\begin{array}{lll} & \text{Variable: } v \in \mathbb{R}^n & \\ \text{s.t.} & v_i^2 = \frac{1}{n} & \forall i = 1, \dots, n \\ & \langle v, d_u \rangle^2 = 1 & \forall u = 1, \dots, m \end{array}}$$

Theorem 4.4. *For both the Gaussian and Boolean settings, there exists a constant $c > 0$ such that for all $\varepsilon > 0$ and $\delta \le c\varepsilon$, for $m \le n^{3/2-\varepsilon}$, w.h.p. there is a feasible degree-n^δ SoS solution for Planted Affine Planes.*

It turns out that the Planted Affine Plane problem introduced above is closely related to the following "Boolean vector in a random subspace" problem, which we call the Planted Boolean Vector problem, introduced by Mohanty–Raghavendra–Xu [MRX20] in the context of studying the performance of SoS on computing the Sherrington–Kirkpatrick Hamiltonian.

The Planted Boolean Vector problem is to certify that a random subspace of $\mathbb{R}^n$ is far from containing a Boolean vector. Specifically, we want to certify an upper bound for

$$\mathsf{OPT}(V) := \frac{1}{n} \max_{b \in \{\pm 1\}^n} b^\intercal \Pi_V b,$$

where V is a uniformly random p-dimensional subspace[1] of $\mathbb{R}^n$, and Π_V is the projector onto V. In brief, the relationship to the Planted Affine Plane problem is that the PAP vector v represents the coefficients on a linear combination for the vector b in the span of a basis of V.

An argument of [MRX20] shows that, when $p \ll n$, w.h.p., $\mathsf{OPT}(V) \approx \frac{2}{\pi}$, whereas they also show that w.h.p. assuming $p \ge n^{0.99}$, there is a degree-4 SoS solution with value $1 - o(1)$. They ask whether or not there is a polynomial time algorithm that can certify a tighter bound; we rule out SoS-based algorithms for a larger regime both in terms of SoS degree and the dimension p of the random subspace.

Theorem 4.5. *There exists a constant $c > 0$ such that, for all $\varepsilon > 0$ and $\delta \le c\varepsilon$, for $p \ge n^{2/3+\varepsilon}$, w.h.p. over V there is a degree-n^δ SoS solution for Planted Boolean Vector of value* 1.

[1] V can be specified by a basis, which consists of p i.i.d. samples from $\mathcal{N}(0, \mathrm{id})$.

4.1.1 Related work

We now summarize the existing work on these problems.

Note that $\mathrm{GOE}(n)$ is a particular kind of Wigner matrix ensemble, thereby satisfying the semicircle law, which in this case establishes that the largest eigenvalue of W is $(2 + \mathrm{o}(1)) \cdot \sqrt{n}$ with probability $1 - \mathrm{o}(1)$. Thus, a trivial spectral bound establishes $\mathsf{OPT}(W) \leq (2 + \mathrm{o}(1)) \cdot n^{3/2}$ with probability $1 - \mathrm{o}(1)$. The natural upper bound of $(2 + \mathrm{o}(1)) \cdot n^{3/2}$ obtained via the spectral norm of W is also the value of the degree-2 SoS relaxation [MS16].

However, the true optimum is known to be smaller by a constant factor. In a foundational work based on a variational argument [Par79], Parisi conjectured that ,

$$\mathop{\mathbb{E}}_{W \sim \mathrm{GOE}(n)} [\mathsf{OPT}(W)] \approx 2 \cdot P^* \cdot n^{3/2},$$

where $P^* \approx 0.7632$ is now referred to as the Parisi constant. Talagrand [Tal06] gave a rigorous proof of Parisi's conjecture.

Degree-4 SoS lower bounds on the Sherrington-Kirkpatrick Hamiltonian problem were proved independently by Mohanty–Raghavendra–Xu [MRX20] and Kunisky–Bandeira [KB19] whereas we prove a much stronger degree-n^{δ} SoS lower bound for some constant $\delta > 0$. Our result is obtained by reducing the Sherrington-Kirkpatrick problem to the "Boolean Vector in a Random Subspace" problem which is equivalent to our new Planted Affine Planes problem on the normal distribution. The reduction from Sherrington-Kirkpatrick problem to the "Boolean Vector in a Random Subspace" is due to Mohanty–Raghavendra–Xu [MRX20]. The results of Mohanty–Raghavendra–Xu [MRX20] and Kunisky–Bandeira [KB19] build on a degree-2 SoS lower bounds of Montanari and Sen [MS16].

Degree-4 SoS lower bounds on the "Boolean Vector in a Random Subspace" problem for $p \geq n^{0.99}$ were proved by Mohanty–Raghavendra–Xu in [MRX20] where this problem was introduced. We improve the dependence on p to $p \geq n^{2/3+\epsilon}$ for any $\epsilon > 0$ and obtain a stronger degree-$n^{c\varepsilon}$ SoS lower bound for some absolute constant $c > 0$.

Regarding the search task for the SK model, Talagrand's result was existential. As it turns out, in a recent breakthrough work, the search problem was shown by Montanari to be algorithmically tractable [Mon21]! That is, there a polynomial-time algorithm that, given $W \sim \mathrm{GOE}(n)$, computes an x achieving close to $\mathsf{OPT}(W)$. In view of our results, the SK model is likely to exhibit a gap between certification and search (as mentioned at the start of Section 3.1.1, such gaps are fundamental assuming $\mathsf{NP} \neq \mathsf{co\text{-}NP}$). See also Section 4.6 for more discussion.

A pair of recent works by Zadik et al [ZSWB21] and Diakonikolas–Kane [DK21] shows that the Planted Affine Planes is actually algorithmically tractable if there is not a small amount of additional noise used to "obscure" the hidden plane. Their algorithm is based on the LLL lattice basis reduction algorithm, an algebraic method that is not captured by sum-of-squares.

4.2 Proof Outline

We now provide a high-level description of our approach. The bulk of our technical contribution lies in the SoS lower bound for the Planted Affine Planes problem, Theorem 4.4. We will show that Planted Affine Planes in the Gaussian setting is equivalent to the Planted Boolean Vector problem in Section 4.5. The final reduction from Sherrington-Kirkpatrick to the Planted Boolean Vector problem is due to Mohanty–Raghavendra–Xu [MRX20]; the reduction is also included in Section 4.5 for completeness.

As a starting point to the PAP lower bound, we employ the general techniques in Chapter 3. We pseudocalibrate against a natural planted distribution to produce a good candidate SoS solution $\widetilde{\mathbb{E}}$, and express the associated moment matrix $\mathcal{M}$ in terms of graph matrices. The precise definition of graph matrices for this problem is given in Section 4.2.1. Recall that to prove the SoS lower bound, we must show that $\widetilde{\mathbb{E}}$ satisfies the problem constraints and is PSD. Because of pseudocalibration, Fact 3.24, $\widetilde{\mathbb{E}}$ automatically satisfies the Boolean constraints “$v_i^2 = \frac{1}{n}$".

The main step of the proof is to prove that $\mathcal{M}$ is PSD with high probability, and an outline of the argument is given in Section 4.2.2. The key difficulty is handling a *null space* in the moment matrix. The null space is caused by the constraints “$\langle v, d_u \rangle^2 = 1$” (in fact, any constrained optimization problem induces a null space for the SoS moment matrix). The outline describes how we “remove” the null space from the moment matrix by getting rid of certain “spider shapes” Once we remove the spiders, we can show that the new matrix is close to the identity matrix (thus the original moment matrix was close to a projection matrix and is PSD). The “closeness” here is proven by bounding the spectral norm of the non-spider shapes. Showing that these shapes have $o(1)$ norm is the most crucial step in the proof. This norm bound argument is proven informally in Section 4.2.3.

Finally, the operator $\widetilde{\mathbb{E}}$ unfortunately does not exactly satisfy the PAP constraints “$\langle v, d_u \rangle^2 = 1$”, it only satisfies them up to a tiny error. We use an interesting and rather generic approach to round $\widetilde{\mathbb{E}}$ to a nearby pseudoexpectation operator $\widetilde{\mathbb{E}}'$ which does exactly satisfy the constraints. The overall technique is described in Section 3.2.1. This somewhat long calculation with graph matrices is omitted, see [GJJ$^+$20, Section 7].

It must be checked that $\widetilde{\mathbb{E}}[1] = 1 + o(1)$. This is equivalent to checking that the *low-degree likelihood ratio* cannot be used to distinguish the planted and random distributions. As will be pointed out in Remark 4.20, w.h.p. in the unnormalized pseudocalibration, $\widetilde{\mathbb{E}}[1] = 1 + \mathrm{o}(1)$ and so the error introduced does not impact the statement of any lemmas.

4.2.1 Graph matrices

In this chapter, we use the following concrete definitions of graph matrices.

The graphs that we study have two types of vertices, circles $\bigcirc$ and squares $\square$. We let $\mathcal{C}_m$ be a set of m circles labeled 1 through m, which we denote by ①, ②, . . . , ⓜ, and let $\mathcal{S}_n$ be a set of n squares labeled 1 through n, which we denote by $\boxed{1}$, $\boxed{2}$, . . . , $\boxed{n}$. We will work with bipartite graphs with edges between circles and squares, which have positive integer labels on the edges.

Definition 4.6 (Proper). *An edge-labeled graph is proper if it has no multiedges and no isolated vertices.*

When the graph is proper, such graphs are in one-to-one correspondence with Fourier characters on the vectors d_u. An edge between $\textcircled{u}$ and $\boxed{i}$ with label l represents $h_l(d_{u,i})$ where $\{h_k\}$ is the Fourier basis for the distribution of a single entry $d_{u,i}$.

$$\text{simple graph with labeled edges} \qquad \iff \qquad \prod_{\substack{\textcircled{u} \in \mathcal{C}_m, \\ \boxed{i} \in \mathcal{S}_n}} h_{l(\textcircled{u}, \boxed{i})}(d_{u,i})$$

In the Gaussian case, $\{h_k\}$ are the (unnormalized, probabilist's) Hermite polynomials, and in the Boolean case, they are just the parity function, represented by

$$h_0(x) = 1, \qquad h_1(x) = x, \qquad h_k(x) = 0 \ \ (k \geq 2)\,.$$

An example of a Fourier polynomial as a graph with labeled edges is given in Fig. 4.1. Unlabeled edges are implicitly labeled 1.

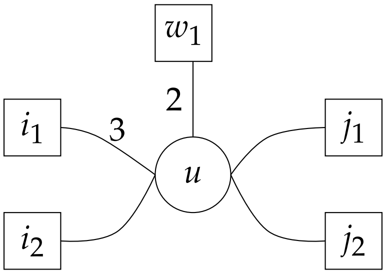


Figure 4.1: The Fourier polynomial $h_3(d_{u,i_1})h_1(d_{u,i_2})h_2(d_{u,w_1})h_1(d_{u,j_1})h_1(d_{u,j_2})$ represented as a graph.

Define the degree of a vertex v, denoted $\deg(v)$, to be the sum of the labels incident to v, and $|E|$ to be the sum of all labels.

The matrices we study have rows and columns indexed by all subsets of $\mathcal{C}_m \cup \mathcal{S}_n$.

Definition 4.7 (Matrix index). *A matrix index is a set A of elements from $\mathcal{C}_m \cup \mathcal{S}_n$.*

We let $A(\boxed{i})$ or $A(\textcircled{u})$ be 0 or 1 to indicate if the vertex is in A.

Definition 4.8 (Ribbons). *A ribbon is an undirected, edge-labeled graph $R = (V(R), E(R), A_R, B_R)$, where $V(R) \subseteq \mathcal{C}_m \cup \mathcal{S}_n$ and A_R, B_R are two matrix indices (possibly not disjoint) with $A_R, B_R \subseteq V(R)$, representing two distinguished sets of vertices. Furthermore, all edges in $E(R)$ go between squares and circles.*

We think of A_R and B_R as being the "left" and "right" sides of R, respectively. We also define the set of "middle vertices" $C_R := V(R) \setminus (A_R \cup B_R)$. If $e \notin E(R)$, then we define its label $l(e) = 0$. We also abuse notation and write $l(\boxed{i}, \textcircled{u})$ instead of $l(\{\boxed{i}, \textcircled{u}\})$.

Definition 4.9 (Matrix for a ribbon)**.** *The matrix M_R has rows and columns indexed by subsets of $\mathcal{C}_m \cup \mathcal{S}_n$, with a single nonzero entry defined by*

$$M_R[I,J] = \begin{cases} \prod\limits_{\substack{e\in E(R),\\ e=\{\boxed{i},\textcircled{u}\}}} h_{l(e)}(d_{u,i}) & I = A_R, J = B_R \\ 0 & \textit{Otherwise} \end{cases}$$

The shape of a ribbon forgets all the vertex labels and retains only the graph structure and the distinguished sets of vertices.

Definition 4.10 (Shape)**.** *Letting S_n permute the square labels and S_m permute the circle labels, a shape is an orbit of ribbons under the action by $S_m \times S_n$. It can be specified by a representative $\alpha = (V(\alpha), E(\alpha), U_\alpha, V_\alpha)$.*

We let $U(\boxed{i})$ and $U(\textcircled{u})$ be either 0 or 1 for whether $\boxed{i}$ or $\textcircled{u}$, respectively, is in U. We'll also use $W_\alpha := V(\alpha) \setminus (U_\alpha \cup V_\alpha)$ to denote the "middle vertices" of the shape.

Remark 4.11. *We will use $\boxed{i}, \boxed{j}, \textcircled{u}, \textcircled{v}, \dots$ for both the vertices of ribbons and the vertices of shapes. In ribbons, i, j, u, v can be either fixed or variable, whereas in shapes, they will always be variable to avoid picking a specific representative of the shape.*

Finally, given a shape α, the graph matrix M_α consists of all Fourier characters for ribbons of shape α.

Definition 4.12 (Graph matrix)**.** *Given a shape α, the graph matrix M_α is*

$$M_\alpha = \sum_{\textit{ribbon } R \textit{ of shape } \alpha} M_R$$

The moment matrix for PAP will turn out to be defined using graph matrices M_α whose left and right sides only have square vertices, and no circles. However, in the course of the analysis we will factor and multiply graph matrices with circle vertices in the left or right.

The spectral norm of a graph matrix is determined, up to logarithmic factors, by relatively simple combinatorial properties of the graph.

Definition 4.13 (Weight)**.** *For a subset $S \subseteq V(\alpha)$, we define the weight*

$$w(S) \; := \; (\textit{\# circles in } S) \cdot \log_n(m) + (\textit{\# squares in } S)\,.$$

Observe that $n^{w(S)} = m^{\textit{\# circles in } S} \cdot n^{\textit{\# squares in } S}$.

We define $S_{\min}$ of a shape α to be the minimum-weight vertex separator of U_α and V_α. Let I_α denote the isolated (degree 0) vertices in W_α. Then whp for all shapes that appear in the analysis (a formal statement is given in Lemma 2.17):

$$\|M_\alpha\| \leq \widetilde{\mathrm{O}}\left(n^{\frac{w(V(\alpha)) - w(S_{\min}) + w(I_\alpha)}{2}}\right).$$

4.2.2 Outline of PSD-ness proof

After performing pseudocalibration, in both settings, we will have essentially the graph matrix decomposition,

$$\mathcal{M} = \sum_{\text{shapes } \alpha} \lambda_\alpha M_\alpha = \sum_{\substack{\text{shapes } \alpha: \\ U_\alpha, V_\alpha \subseteq \mathcal{S}_n, \\ \deg(\boxed{i}) + U(\boxed{i}) + V(\boxed{i}) \text{ even}, \\ \deg(\textcircled{u}) \text{ even}}} \frac{1}{n^{\frac{|U_\alpha|+|V_\alpha|}{2}}} \cdot \left(\prod_{\textcircled{u} \in V(\alpha)} h_{\deg(\textcircled{u})}(1) \right) \cdot \frac{M_\alpha}{n^{|E(\alpha)|/2}}. \tag{4.2}$$

Here $h_k(1)$ is in both settings the k-th Hermite polynomial, evaluated on 1. To show that the matrix $\mathcal{M}$ is PSD, we need to study the graph matrices that appear with nonzero coefficients in the decomposition.

First attempts. We might try to use the techniques in Section 3.3, for example showing that $\mathbb{E}\,\mathcal{M}$ is dominant, or using the approximate PSD decomposition. A fundamental obstruction to using these approaches is that the nullspace of $\mathcal{M}$ is nontrivial, whereas these arguments show a positive lower bound on the minimum eigenvalue of $\mathcal{M}$. First we explain how the naive approach fails, then we explain why the nullspace exists and how we get around it.

Recall that a shape $\alpha = (V(\alpha), E(\alpha), U_\alpha, V_\alpha)$ is trivial if $V(\alpha) = U_\alpha = V_\alpha$ and $E(\alpha) = \emptyset$. The trivial shapes in the expression for $\mathcal{M}$ contribute (scaled) identity matrices $\frac{1}{n^{k/2}}\mathrm{Id}$ on the degree-k block of the moment matrix. If we could bound $|\lambda_\alpha| \|M_\alpha\| \ll \frac{1}{n^{\frac{|U_\alpha|+|V_\alpha|}{2}}}$ for all non-trivial graph matrices in $\mathcal{M}$, then the identity matrices are dominant and $\mathcal{M}$ is PSD.

For several shapes this strategy is indeed viable. To illustrate, let's consider one such shape α depicted in Fig. 4.2.

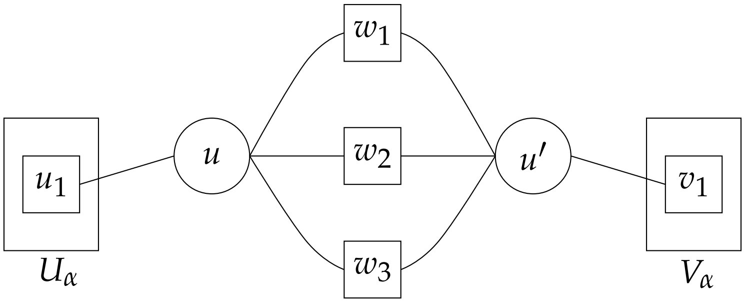


Figure 4.2: Picture of basic non-spider shape α.

This graph matrix has $|\lambda_\alpha| = \Theta(\frac{1}{n^5})$. Using the graph matrix norm bounds, with high probability the norm of this graph matrix is $\tilde{O}(n^2 m)$: there are four square vertices and two circle vertices which are not in the minimum vertex separator. Thus, for this shape α, with high probability $|\lambda_\alpha| \|M_\alpha\|$ is $\tilde{O}\left(\frac{m}{n^3}\right)$ and thus $\lambda_\alpha M_\alpha \preceq \frac{1}{n}\mathrm{Id}$ (which is the multiple of the identity appearing in the corresponding block).

Unfortunately, some shapes α that appear in the decomposition have $\|\lambda_\alpha M_\alpha\|$ too large to be charged against the identity matrices. These are shapes with a certain substructure whose norms cannot be handled by the preceding argument, and which we denote *spiders*. The following graph depicts one such *spider* shape (and also motivates this terminology):

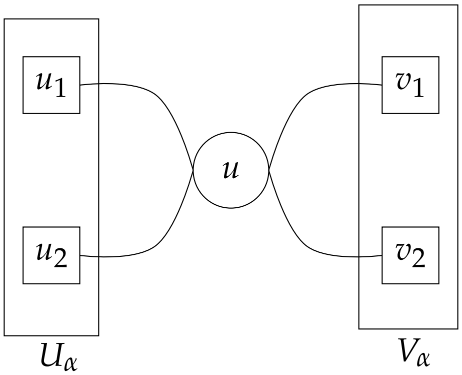


Figure 4.3: Picture of basic spider shape α.

The norm $\|\lambda_\alpha M_\alpha\|$ of this graph is $\widetilde{\mathrm{O}}(\frac{1}{n^2})$, as can be easily estimated through the norm bounds (the coefficient is $\lambda_\alpha = \frac{-2}{n^4}$, the minimum vertex separator is $\textcircled{u}$, and there are no isolated vertices). This is too large to bound against $\frac{1}{n^2}\mathrm{Id}$, which is the identity matrix on this spider's block.[2]

Existence of the null space. The kernel of the matrix $\mathcal{M}$ is nontrivial as a consequence of satisfying the PAP constraints "$\langle v, d_u \rangle^2 = 1$". Consider the two shapes in Fig. 4.4, β_1 and β_2 (take note of the label 2 on the edge in β_2).

We claim that if $\mathcal{M}$ is any moment matrix satisfying the PAP constraints, then every column of the matrix $2M_{\beta_1} + \frac{1}{n}M_{\beta_2}$ is in the null space of $\mathcal{M}$. There are m nonzero columns indexed by assignments to V, which can be a single circle $\textcircled{1}, \textcircled{2}, \dots, \textcircled{m}$. The nonzero rows are $\emptyset$ in β_2 and $\{\boxed{i}, \boxed{j}\}$ for $i \neq j$ in β_1. Fixing $I \subseteq [n]$, entry $(I, \textcircled{u})$ of the

[2] There is a related obstruction to using the approximate PSD decomposition in addition to $\mathcal{M}$'s nontrivial nullspace. To use the approximate PSD decomposition, we need $\lambda_{\sigma\circ\sigma^\intercal}$ for each approximately PSD shape $\sigma \circ \sigma^\intercal$. However, the spider shape is of this form, but has $\lambda_{\sigma\circ\sigma^\intercal} < 0$.

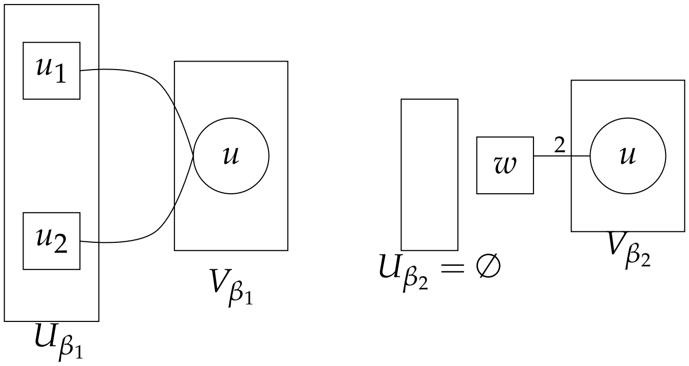


Figure 4.4: Picture of shapes β_1 and β_2.

product matrix $\mathcal{M}(2M_{\beta_1} + \frac{1}{n}M_{\beta_2})$ is

$$\begin{aligned}
&2\sum_{i<j} \widetilde{\mathbb{E}}[v^I v_i v_j] \cdot d_{ui} d_{uj} + \frac{1}{n}\widetilde{\mathbb{E}}[v^I] \cdot \sum_i (d_{ui}^2 - 1) \\
&= 2\sum_{i<j} \widetilde{\mathbb{E}}[v^I v_i v_j] \cdot d_{ui} d_{uj} + \widetilde{\mathbb{E}}[v^I v_i^2] \cdot \sum_i d_{ui}^2 - \widetilde{\mathbb{E}}[v^I] \qquad (\widetilde{\mathbb{E}} \text{ satisfies } \text{“}v_i^2 = \frac{1}{n}\text{”}) \\
&= \sum_{i,j} \widetilde{\mathbb{E}}[v^I v_i v_j] d_{ui} d_{uj} - \widetilde{\mathbb{E}}[v^I] \\
&= \widetilde{\mathbb{E}}[v^I(\langle v, d_u\rangle^2 - 1)] \\
&= 0 \qquad (\widetilde{\mathbb{E}} \text{ satisfies “} \langle v, d_u\rangle^2 = 1\text{”})
\end{aligned}$$

In words, the constraint “$\langle v, d_u\rangle^2 = 1$” creates a shape $2\beta_1 + \frac{1}{n}\beta_2$ that lies in the null space of the moment matrix.

Handling the null space. To skirt the spiders, we restrict ourselves to vectors $x \perp \mathrm{Null}(M)$. Clearly it is sufficient to show $x^\intercal \mathcal{M} x \geq 0$ to prove $\mathcal{M} \succeq 0$.

We claim the basic spider α in Fig. 4.3 satisfies $x^\intercal M_\alpha \approx 0$. We can approximately factor the spider α across its central vertex, and when we do so, the shape β_1 appears on the left side. Therefore $M_\alpha \approx M_{\beta_1} M_{\beta_1}^\intercal \approx (M_{\beta_1} + \frac{1}{2n}M_{\beta_2})M_{\beta_1}^\intercal$. The columns of the matrix $M_{\beta_1} + \frac{1}{2n}M_{\beta_2}$ are in the null space of $\mathcal{M}$, so for $x \perp \mathrm{Null}(\mathcal{M})$ we have $x^\intercal M_\alpha \approx 0$ as claimed.

The graphical substructure present in a spider implies that the spider is close to the zero matrix in $\mathrm{Null}(\mathcal{M})^\perp$. Because of this, we can almost freely add and subtract M_α for spiders α while preserving the action of $\mathcal{M}$ on $\mathrm{Null}(\mathcal{M})^\perp$. Our strategy is to “kill” the spiders by subtracting off $\lambda_\alpha \cdot M_\alpha$ for each spider α. Doing this for all spiders, we produce a matrix whose action is equivalent on $\mathrm{Null}(\mathcal{M})^\perp$, and which is dominated by the identity matrix by virtue of the fact that it has no spiders, showing that $\mathcal{M}$ is PSD.

More formally, handling the approximate equalities, for a spider α we are able to find

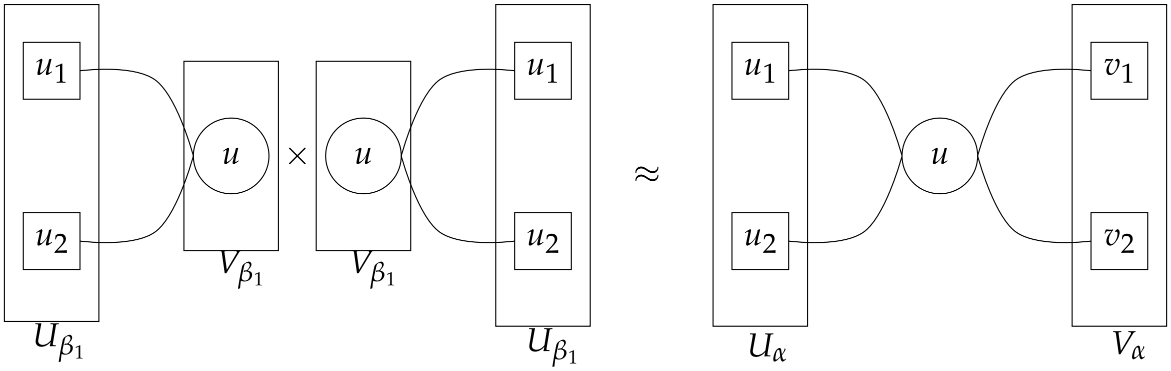


Figure 4.5: Approximation $\beta_1 \times \beta_1^\intercal \approx \alpha$.

coefficients c_β for some non-dominant intersection terms β so that all columns of the matrix

$$A = M_\alpha + \sum_\beta c_\beta M_\beta$$

are in $\mathrm{Null}(\mathcal{M})$. We then observe the following fact:

Fact 4.14. *If $x \perp \mathrm{Null}(\mathcal{M})$ and $\mathcal{M}A = 0$, then for all matrices B, $x^\intercal(AB + \mathcal{M})x = x^\intercal(B^\intercal A^\intercal + \mathcal{M})x = x^\intercal \mathcal{M} x$.*

Using the fact, we can freely add multiples of A to $\mathcal{M}$ without changing the action of $\mathcal{M}$ on $\mathrm{Null}(\mathcal{M})^\perp$. A judicious choice is to subtract $\lambda_\alpha A$ which will "kill" the spider from $\mathcal{M}$.

The intersection terms β may themselves be spiders (though they will always have fewer square vertices than α). Thus we must recursively kill these spiders, until there are no spiders remaining in the decomposition of $\mathcal{M}$. With $\equiv$ denoting equality of the matrix on $\mathrm{Null}(\mathcal{M})^\intercal$, here is informally how the recursive process unfolds:

$$\begin{aligned}
\mathcal{M} &= \sum_{\text{non-spiders } \alpha} \lambda_\alpha M_\alpha + \sum_{\text{spiders } \alpha} \lambda_\alpha M_\alpha \\
&\equiv \sum_{\text{non-spiders } \alpha} \lambda_\alpha M_\alpha - \sum_{\text{spiders } \alpha} \sum_{\substack{\text{intersection} \\ \text{terms } \beta}} \lambda_\alpha c_\beta M_\beta \\
&= \sum_{\text{non-spiders } \alpha} \lambda_\alpha M_\alpha - \sum_{\text{spiders } \alpha} \sum_{\substack{\text{intersection} \\ \text{non-spiders } \beta}} \lambda_\alpha c_\beta M_\beta - \sum_{\text{spiders } \alpha} \sum_{\substack{\text{intersection} \\ \text{spiders } \beta}} \lambda_\alpha c_\beta M_\beta \\
&\equiv \sum_{\text{non-spiders } \alpha} \lambda_\alpha M_\alpha - \sum_{\text{spiders } \alpha} \sum_{\substack{\text{intersection} \\ \text{non-spiders } \beta}} \lambda_\alpha c_\beta M_\beta + \sum_{\text{spiders } \alpha} \sum_{\substack{\text{intersection} \\ \text{spiders } \beta}} \sum_{\substack{\text{intersection} \\ \text{terms } \beta'}} \lambda_\alpha c_\beta c_{\beta'} M_{\beta'} \\
&\quad\vdots \\
&\equiv \sum_{\text{non-spiders } \alpha} \lambda_\alpha M_\alpha \pm \sum_{\text{spiders } \alpha} \sum_{\substack{\text{intersection} \\ \text{spiders } \beta_1}} \sum_{\substack{\text{intersection} \\ \text{spiders } \beta_2}} \cdots \sum_{\substack{\text{intersection} \\ \text{spiders } \beta_\ell}} \sum_{\substack{\text{terminal} \\ \text{non-spiders } \beta}} \lambda_\alpha c_{\beta_1} \cdots c_{\beta_\ell} M_\beta .
\end{aligned}$$

The total coefficient on any terminal non-spider β will be relatively small, but this must be proven combinatorially.

4.2.3 Informal sketch of key charging lemmas

The critical part of the proof outline is to show that the two types of "terminal" shapes can be charged to the identity matrices: non-spiders originally in $\mathcal{M}$, and non-spiders that result from killing the spiders. Using combinatorial arguments with the norm bounds, we prove these two lemmas here.

Non-spiders are negligible. We prove that non-spiders initially in the moment matrix are dominated in norm by the identity matrix. We point out that this charging argument critically relies on the assumption $m \leq n^{3/2-\varepsilon}$.

Definition 4.15 (Spider). *A left spider is a proper shape $\alpha = (V(\alpha), E(\alpha), U_\alpha, V_\alpha)$ with the property that there exist two distinct square vertices $\boxed{i}, \boxed{j} \in U_\alpha$ of degree 1 and a circle vertex $\textcircled{u} \in V(\alpha)$ such that $E(\alpha)$ contains the edges $(\boxed{i}, \textcircled{u})$ and $(\boxed{j}, \textcircled{u})$ (these are necessarily the only edges incident to $\boxed{i}$ and $\boxed{j}$).*

A right spider is the transpose of a left spider. A spider is a left or right spider.

The vertices $\boxed{i}$ and $\boxed{j}$ are called the *end vertices* of α. Because of degree parity, the end vertices must lie in $U_\alpha \setminus (U_\alpha \cap V_\alpha)$.

Remark 4.16. *A spider can have many pairs of end vertices. For each possible spider shape, we single out a pair of end vertices, so that in what follows we can discuss "the" end vertices of the spider.*

Lemma 4.17 (Non-spiders are negligible). *If proper shape α is not a spider and α is non-trivial, then $|\lambda_\alpha| \|M_\alpha\| \ll \frac{1}{n^{\frac{|U_\alpha|+|V_\alpha|}{2}}}$.*

Proof. Plugging in the pseudocalibrated λ_α from Eq. (4.2), the shapes α with $\lambda_\alpha \neq 0$ have the structural properties:

(i) The left and right sides contain only squares: $U_\alpha \cup V_\alpha \subseteq \mathcal{S}_n$,

(ii) Degree parity constraints: $\deg(\boxed{i}) + U(\boxed{i}) + V(\boxed{i})$ even, $\deg(\textcircled{u})$ even.

Plugging in the norm bounds:

$$|\lambda_\alpha| \|M_\alpha\| \leq \frac{1}{n^{\frac{|U_\alpha|+|V_\alpha|}{2}}} \cdot \left| \prod_{\textcircled{u} \in V(\alpha)} h_{\deg(\textcircled{u})}(1) \right| \cdot \frac{1}{n^{|E(\alpha)|/2}} \cdot n^{\frac{w(V(\alpha)) - w(S_{\min})}{2}}.$$

Since circles have even degree, $h_2(1) = 0$, and no circle can have degree 0 in a proper shape such that $(U_\alpha \cup V_\alpha) \cap \mathcal{C}_m = \emptyset$, we conclude a third structural property:

(iii) $\deg(\textcircled{u}) \geq 4$.

This is the key property we will need of $h_{\deg(\textcircled{u})}(1)$; this coefficient is otherwise upper bounded by Proposition 4.45:

$$\left| \prod_{\textcircled{u} \in V(\alpha)} h_{\deg(\textcircled{u})}(1) \right| \leq |E(\alpha)|^{C|E(\alpha)|}$$

for some constant C. Since $|E(\alpha)|^C \ll \sqrt{n}$, this may be incorporated into $\frac{1}{n^{|E(\alpha)|/2}}$ with negligible loss.

Therefore it suffices to show for non-trivial, non-spider shapes α satisfying structural properties (i), (ii), (iii):

$$\frac{1}{n^{|E(\alpha)|/2}} n^{\frac{w(V(\alpha))-w(S_{\min})}{2}} \ll 1.$$

The idea behind the proof is as follows. Each square vertex which is not in the minimum vertex separator contributes $\sqrt{n}$ to the norm bound while each circle vertex which is not in the minimum vertex separator contributes $\sqrt{m}$. To compensate for this, we will try and take the factor of $\frac{1}{\sqrt{n}}$ from each edge and distribute it among its two endpoints so that each square vertex which is not in the minimum vertex separator is assigned a factor of $\frac{1}{\sqrt{n}}$ or smaller and each circle vertex which is not in the minimum vertex separator is assigned a factor of $\frac{1}{\sqrt{m}}$ or smaller.

Remark 4.18. *Instead of using the minimum vertex separator, we will actually use a set S of square vertices such that $w(S) \leq w(S_{\min})$. For details, see the actual distribution scheme below.*

Remark 4.19. *We will leave $\frac{1}{n^{\Omega(\varepsilon)}}$ unassigned per edge, turning $\leq$ into $\ll$ for non-trivial shapes.*

To motivate the distribution scheme which we use, we first give two attempts which don't quite work. For simplicity, for these first two attempts we assume that $U_\alpha \cap V_\alpha = \emptyset$ as vertices in $U_\alpha \cap V_\alpha$ can essentially be ignored (Section 2.3).

> Attempt 1: Take each edge and assign a factor of $\frac{1}{\sqrt[4]{n}}$ to its square endpoint and a factor of $\frac{1}{\sqrt[8]{m}}$ to its circle endpoint.
>
> With this distribution scheme, since each circle vertex has degree at least 4, each circle vertex is assigned a factor of $\frac{1}{\sqrt{m}}$ or smaller. Since each square vertex in W_α has degree at least 2, each square vertex in W_α is assigned a factor of $\frac{1}{\sqrt{n}}$ or smaller. However, square vertices in $U_\alpha \cup V_\alpha$ may only have degree 1 in which case they are assigned a factor of $\frac{1}{\sqrt[4]{n}}$ which is not small enough.
>
> **Remark 4.20.** *For analyzing $\widetilde{\mathbb{E}}[1]$, this first attempt works as $U_\alpha = V_\alpha = \emptyset$. Thus, as long as $m \leq n^{2-\epsilon}$, with high probability $\widetilde{\mathbb{E}}[1] = 1 \pm o(1)$.*

To fix this issue, we can have all of the edges which are incident to a square vertex in $U_\alpha \cup V_\alpha$ give their entire factor of $\frac{1}{\sqrt{n}}$ to the square vertex.

Attempt 2: For each edge which is between a square vertex in $U_\alpha \cup V_\alpha$ and a circle vertex, we assign a factor of $\frac{1}{\sqrt{n}}$ to the square vertex and nothing to the circle vertex. For all other edges, we assign a factor of $\frac{1}{\sqrt[4]{n}}$ to its square endpoint and a factor of $\frac{1}{\sqrt[6]{m}}$ to its circle endpoint (which we can do because $m \leq n^{\frac{3}{2}-\epsilon}$).

With this distribution scheme, each square vertex is assigned a factor of $\frac{1}{\sqrt{n}}$. Since α is not a spider, no circle vertex is adjacent to two vertices in U_α or V_α. Thus, any circle vertex which is not adjacent to both a square vertex in U_α and a square vertex in V_α must be adjacent to at least 3 square vertices in W_α and is thus assigned a factor of $\frac{1}{\sqrt{m}}$ or smaller. However, we can have circle vertices which are adjacent to both a square vertex in U_α and a square vertex in V_α. These circle vertices may be assigned a factor of $\frac{1}{\sqrt[3]{m}}$, which is not small enough.

To fix this, observe that whenever we have a circle vertex which is adjacent to both a square vertex in U_α and a square vertex in V_α, this gives a path of length 2 from U_α to V_α. Any vertex separator must contain one of the vertices in this path, so we can put one of these two square vertices in S and not assign it a factor of $\frac{1}{\sqrt{n}}$.

Actual distribution scheme: Based on these observations, we use the following distribution scheme (see Fig. 4.6 for an example). Here we are no longer assuming that $U_\alpha \cap V_\alpha$ is empty.

1. Choose a set of square vertices $S \subseteq U_\alpha \cup V_\alpha$ as follows. Start with $S = U_\alpha \cap V_\alpha$. Whenever we have a circle vertex which is adjacent to both a degree-1 square vertex in $U_\alpha \setminus V_\alpha$ and a degree-1 square vertex in $V_\alpha \setminus U_\alpha$, put one of these two square vertices in S (this choice is arbitrary). Observe that $w(S) \leq w(S_{\min})$.

2. For each edge which is incident to a square vertex in S, assign a factor of $\frac{1}{\sqrt[3]{m}}$ to its circle endpoint and nothing to this square.

3. For each edge which is incident to a degree-1 square vertex in $(U_\alpha \cup V_\alpha) \setminus S$, assign a factor of $\frac{1}{\sqrt{n}}$ to the square vertex and nothing to the circle vertex.

4. For all other edges, assign a factor of $\frac{1}{\sqrt[4]{n}}$ to its square endpoint and a factor of $\frac{1}{\sqrt[6]{m}}$ to its circle endpoint.

Now each square vertex which is not in S is assigned a factor of $\frac{1}{\sqrt{n}}$ or smaller. Since α is not a spider, all circle vertices are incident to at most one edge from case 3 and are assigned a factor of $\frac{1}{\sqrt{m}}$ or smaller.

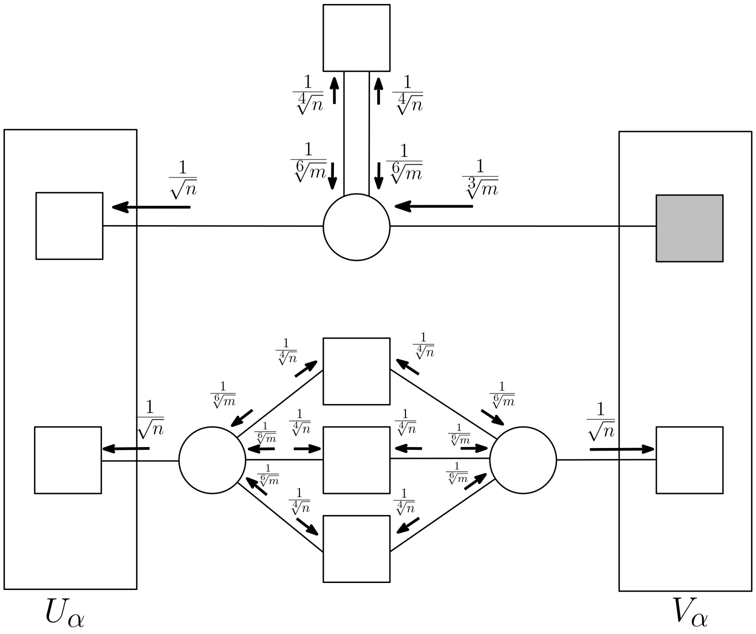


Figure 4.6: Allocation of $\frac{1}{\sqrt{n}}$ from each edge to vertices. S consists of the gray square. Observe that every circle receives a factor of at least $\frac{1}{\sqrt{m}}$ and every square not in S receives a factor of at least $\frac{1}{\sqrt{n}}$.

For each edge, $\frac{1}{n^{\Omega(\varepsilon)}}$ is unassigned because $m \leq n^{3/2-\varepsilon}$, with the exception of case 3 in which the entire factor of $\frac{1}{\sqrt{n}}$ is assigned. This case is less than one quarter of the edges of α, since the other edges incident to the circle must not be case 3, thus we can still have $\frac{1}{n^{\Omega(\varepsilon)}}$ unassigned per edge on average. □

Remark 4.21. *The non-spider depicted in Fig. 4.7 is an obstruction to improving the assumption on m from $n^{3/2-\varepsilon}$ to $n^{2-\varepsilon}$. We have $|\lambda_\alpha| \|M_\alpha\| = \widetilde{O}\left(\frac{m^2}{n^5}\right)$ using the norm bounds. The multiple of the identity appearing on this block is $\frac{1}{n^2}\mathrm{Id}$, which is only larger if $m \ll n^{3/2}$.*

Terminal non-spiders are negligible. The non-spiders β that arise when killing a spider α are certain intersection terms, and they do not need to have all the structural properties that we utilized in the previous proof. In particular, the shapes may be improper, which when linearized may have circle vertices of degree 2 or isolated vertices.

To handle the potentially larger norms, we will use that the coefficient of a new non-spider term β is at most λ_α for the spider term α. Since α has more vertices/edges than β, the extra factors of $\frac{1}{n}$ in λ_α are used to cancel out any increase in norm.

We will not formally define the set of intersection terms that arise until Section 4.4. It suffices to note that they are improper shapes β that are intersections of α with the

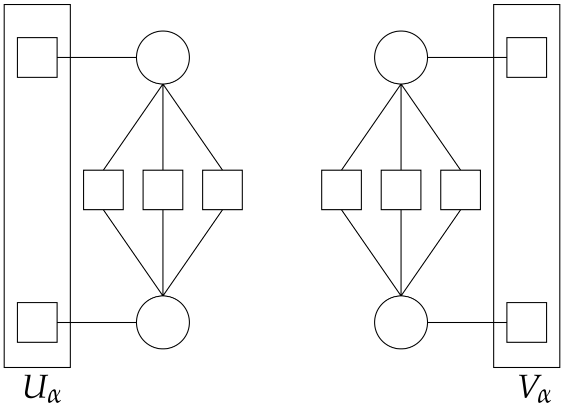


Figure 4.7

following structural properties (properties (i), (ii), (iii) are the same as in the previous proof):

(i) $U_\beta \cup V_\beta \subseteq \mathcal{S}_n$

(ii) Degree parity constraints: $\deg(\boxed{i}) + U(\boxed{i}) + V(\boxed{i})$ even, $\deg(\textcircled{u})$ even.

(iii) $\deg(\textcircled{u}) \geq 4$

(iv) Circles never intersect.

(v) If $\boxed{i}$ intersected, then $\boxed{i}$ was an end vertex at some point.

Lemma 4.22 (Terminal non-spiders are negligible)**.** *If β is an improper non-spider which is an intersection of α, and β satisfies properties (i)-(v), then*

$$|\lambda_\alpha| \left\| M_\beta \right\| \ll \frac{1}{n^{\frac{|U_\beta|+|V_\beta|}{2}}}\,.$$

Proof. Plugging in λ_α and the norm bounds,

$$|\lambda_\alpha| \left\| M_\beta \right\| \leq \frac{1}{n^{\frac{|U_\alpha|+|V_\alpha|}{2}}} \cdot \left| \prod_{\textcircled{u} \in V(\alpha)} h_{\deg(\textcircled{u})}(1) \right| \cdot \frac{1}{n^{|E(\alpha)|/2}} \cdot n^{\frac{w(V(\beta)) - w(S_{\min}) + w(I_\beta)}{2}}\,.$$

As in the previous proof, the Hermite terms may be upper bounded, and it suffices to prove:

$$\frac{1}{n^{\frac{|U_\alpha|+|V_\alpha|-|U_\beta|-|V_\beta|}{2}}} \cdot \frac{1}{n^{|E(\alpha)|/2}} \cdot n^{\frac{w(V(\beta)) - w(S_{\min}) + w(I_\beta)}{2}} \ll 1\,.$$

We will allocate factors to vertices in such a way that every isolated square vertex is allocated at least $\frac{1}{n}$, every isolated circle vertex is allocated at least $\frac{1}{m}$, every square vertex not in $S_{\min}$ is allocated at least $\frac{1}{\sqrt{n}}$, and every circle vertex not in $S_{\min}$ is allocated at least $\frac{1}{\sqrt{m}}$.

In addition to $\frac{1}{\sqrt{n}}$ per edge, we may also allocate $\frac{1}{\sqrt{n}}$ each time that a vertex disappears from U_α or V_α. These factors should be allocated as follows. $|U_\alpha| + |V_\alpha|$ shrinks if two square vertices in U_α intersect, or two square vertices in V_α intersect. When this happens, the new vertex is no longer in U_α (resp. V_α), so $|U_\alpha| + |V_\alpha|$ shrinks by two. Allocate two factors of $\frac{1}{\sqrt{n}}$ to the intersected square vertex.

To allocate the edge factors, the charging strategy we use is:

1. If the edge has a parallel edge (there is a multiedge between these two vertices), assign a factor of $\frac{1}{\sqrt[4]{m}}$ to its circle endpoint and nothing to its square endpoint.

2. For the remaining edges, we use the same charging scheme as the non-spider charging scheme for β with all multiedges deleted.

If $\textcircled{u}$ is isolated in β with all multiedges deleted, all incident edges in β (of which there are at least 4) are multiedges, therefore this vertex is allocated at least $\frac{1}{m}$.

If $\boxed{i}$ is isolated in β with all multiedges deleted, the incident edges in β are multiedges, and hence $\boxed{i}$ must have intersected (since circles don't intersect by structural property (iv)). Therefore, by property (v), $\boxed{i}$ was an end vertex at some point, and therefore in U_α or V_α. In order to become isolated, $\boxed{i}$ moved out of U_α or V_α at some point; at this point it was allocated a factor of $\frac{1}{n}$.

By the previous charging argument for non-spiders, the non-isolated vertices are allocated sufficiently. The only edge case is circle vertices of degree 2. Since these vertices are allocated $\frac{1}{\sqrt[4]{m}}$ per incident multiedge, of which there are at least 2, these vertices are already allocated $\frac{1}{\sqrt{m}}$. □

Remark 4.23. *Surprisingly the scaling factor of $\frac{1}{\sqrt{n}}$ is helpful here, as we allocate it to isolated square vertices. See the discussion in [GJJ$^+$20, Appendix D].*

Remark 4.24. *This part of the charging argument works under the weaker assumption $m \leq n^{2-\varepsilon}$.*

4.3 Pseudocalibration

We formally define the random and the planted distributions for the Planted Affine Planes problem in the Gaussian and Boolean settings. These two (families of) distributions are required by the pseudocalibration machinery in order to define a candidate pseudoexpectation operator $\widetilde{\mathbb{E}}$. For the Gaussian setting, we have the following distributions.

Definition 4.25 (Gaussian PAP distributions)**.** *The Gaussian PAP distributions are as follows.*

1. *(Random distribution) m i.i.d. vectors $d_u \sim \mathcal{N}(0, \mathrm{Id}_n)$.*

2. *(Planted distribution) A vector v is sampled uniformly from $\left\{\pm\frac{1}{\sqrt{n}}\right\}^n$, as well as signs $b_u \in_{\mathrm{R}} \{\pm 1\}$, and m vectors d_u are drawn from $\mathcal{N}(0, \mathrm{Id}_n)$ conditioned on $\langle d_u, v\rangle = b_u$.*

For the Boolean setting, we have the following distributions.

Definition 4.26 (Boolean PAP distributions)**.** *The Boolean PAP distributions are as follows.*

1. *(Random distribution) m i.i.d. vectors $d_u \in_{\mathrm{R}} \{-1, +1\}^n$.*
2. *(Planted distribution) A vector v is sampled uniformly from $\left\{\pm\frac{1}{\sqrt{n}}\right\}^n$, as well as signs $b_u \in_{\mathrm{R}} \{\pm 1\}$, and m vectors d_u are drawn from $\{\pm 1\}^n$ conditioned on $\langle d_u, v\rangle = b_u$.*

For the pseudocalibration we truncate to only Fourier coefficients of size at most n^τ.

4.3.1 Gaussian setting pseudocalibration

We start by computing the pseudocalibration for the Gaussian setting. Here the natural choice of Fourier basis is the Hermite polynomials. Let $\alpha \in (\mathbb{N}^n)^m$ denote a Hermite polynomial index. Define $\alpha! := \prod_{u,i} \alpha_{u,i}!$ and $|\alpha| := \sum_{u,i} \alpha_{u,i}$ and $|\alpha_u| := \sum_i \alpha_{u,i}$. We let $h_\alpha(d_1, \dots, d_m)$ denote an unnormalized Hermite polynomial, so that $h_\alpha / \sqrt{\alpha!}$ forms an orthonormal basis for polynomials in the entries of the vectors $d_1, \dots, d_m$, under the inner product $\langle p, q\rangle = \mathbb{E}_{d_1,\dots,d_m \sim \mathcal{N}(0,\mathrm{Id})}[p \cdot q]$.

We can view α as an $m \times n$ matrix of natural numbers, and with this view we also define $\alpha^\mathsf{T} \in (\mathbb{N}^m)^n$.

Lemma 4.27. *For any $I \subseteq [n]$, the pseudocalibration value is*

$$\widetilde{\mathbb{E}}\, v^I = \sum_{\substack{\alpha : |\alpha| \le n^\tau, \\ |\alpha_u| \text{ even}, \\ |(\alpha^\mathsf{T})_i| \equiv I_i \pmod 2}} \left(\prod_{u=1}^m h_{|\alpha_u|}(1)\right) \cdot \frac{1}{n^{|I|/2 + |\alpha|/2}} \cdot \frac{h_\alpha(d_1, \dots, d_m)}{\alpha!}.$$

In words, the nonzero Fourier coefficients are those which have even row sums, and whose column sums match the parity of I.

Proof. The truncated pseudocalibrated value is defined to be

$$\widetilde{\mathbb{E}}\, v^I = \sum_{\alpha : |\alpha| \le n^\tau} \frac{h_\alpha(d_1, \dots, d_m)}{\alpha!} \cdot \mathop{\mathbb{E}}_{\mathrm{pl}}[h_\alpha(d_1, \dots, d_m) \cdot v^I]$$

So we set about to compute the planted moments. For this computation, the following lemma is crucial. Here, we give a short proof of this lemma using generating functions. For a combinatorial proof, see [GJJ+20, Appendix C].

Lemma 4.28. *Let $\alpha \in \mathbb{N}^n$. When v is fixed and b is fixed (not necessarily +1 or -1) and $d \sim N(0, I)$ conditioned on $\langle v, d\rangle = b\|v\|$,*

$$\mathop{\mathbb{E}}_{d}[h_\alpha(d)] = \frac{v^\alpha}{\|v\|^{|\alpha|}} \cdot h_{|\alpha|}(b).$$

Proof. It suffices to prove the claim when $\|v\| = 1$ since the left-hand side is independent of $\|v\|$. Express $d = bv + (I - vv^\intercal)x$ where $x \sim N(0, \mathrm{Id})$ is a standard normal variable. Now we want

$$\mathop{\mathbb{E}}_{x \sim N(0,\mathrm{Id})} h_\alpha\left(bv + (I - vv^\intercal)x\right).$$

The Hermite polynomial generating function is

$$\sum_{\alpha \in \mathbb{N}^n} \mathop{\mathbb{E}}_{x \sim \mathcal{N}(0,\mathrm{Id})} h_\alpha\left(bv + (I - vv^\intercal)x\right) \frac{t^\alpha}{\alpha!} = \mathop{\mathbb{E}}_{x} \exp\left(\langle bv + (I - vv^\intercal)x, t\rangle - \frac{\|t\|_2^2}{2}\right)$$

$$= \int_{\mathbb{R}^n} \frac{1}{(2\pi)^{\frac{n}{2}}} \cdot \exp\left(\langle bv + (I - vv^\intercal)x, t\rangle - \frac{\|t\|_2^2}{2} - \frac{\|x\|_2^2}{2}\right) dx.$$

Completing the square,

$$\begin{aligned} &= \int_{\mathbb{R}^n} \frac{1}{(2\pi)^{\frac{n}{2}}} \cdot \exp\left(\langle bv, t\rangle - \frac{\langle v, t\rangle^2}{2} - \frac{1}{2} \cdot \|x - (t - \langle v, t\rangle\, v)\|_2^2\right) dx \\ &= \exp\left(\langle bv, t\rangle - \frac{\langle v, t\rangle^2}{2}\right) \\ &= \exp\left(b\,\langle v, t\rangle - \frac{1}{2} \cdot \langle v, t\rangle^2\right). \end{aligned}$$

How can we Taylor expand this in terms of t? The Taylor expansion of $\exp(by - \frac{y^2}{2})$ is $\sum_{i=0}^\infty h_i(b) \frac{y^i}{i!}$. That is, the i-th derivative in y of $\exp(by - \frac{y^2}{2})$, evaluated at 0, is $h_i(b)$. Using the chain rule with $y = \langle v, t\rangle$, the α-derivative in t of our expression, evaluated at 0, is $v^\alpha \cdot h_{|\alpha|}(b)$. This is the expression we wanted when $\|v\| = 1$, and along with the aforementioned remark about homogeneity in $\|v\|$ this completes the proof. □

Now we can finish the calculation. To compute $\mathbb{E}_{\mathrm{pl}}[h_\alpha(d_1, \ldots, d_m) \cdot v^I]$, marginalize v

and the b_u and factor the conditionally independent b_u and d_u.

$$\begin{aligned}
\mathop{\mathbb{E}}_{\text{pl}}[h_\alpha(d_1,\dots,d_m)v^I] &= \mathop{\mathbb{E}}_{v,b_u} v^I \prod_{u=1}^{m} \mathop{\mathbb{E}}_{d}\left[h_{\alpha_u}(d_u) \mid v, b_u\right] \\
&= \mathop{\mathbb{E}}_{v,b_u} v^I \cdot \prod_{u=1}^{m} \frac{v^{\alpha_u}}{\|v\|^{|\alpha_u|}} \cdot h_{|\alpha_u|}(b_u) && \text{(Lemma 4.28)} \\
&= \left(\mathop{\mathbb{E}}_{v} \frac{v^{I+\sum_{u=1}^{m}\alpha_u}}{\|v\|^{\sum_{u=1}^{m}|\alpha_u|}}\right) \cdot \left(\prod_{u=1}^{m} \mathop{\mathbb{E}}_{b_u} h_{|\alpha_u|}(b_u)\right)
\end{aligned}$$

The Hermite polynomial expectations will be zero in expectation over b_u if the degree is odd, and otherwise b_u is raised to an even power and can be replaced by 1. This requires that $|\alpha_u|$ is even for all u. The norm $\|v\|$ is constantly 1 and can be dropped. The numerator will be $\frac{1}{n^{|I|/2+|\alpha|/2}}$ if the parity of every $|(\alpha^\mathsf{T})_i|$ matches I_i, and 0 otherwise. This completes the pseudocalibration calculation. □

We can now write $\mathcal{M}$ in terms of graph matrices.

Definition 4.29. *Let $\mathcal{L}$ be the set of all proper shapes α with the following properties*

- *U_α and V_α only contain square vertices and $|U_\alpha|, |V_\alpha| \le n^\delta$*
- *W_α has no degree 0 vertices*
- $\deg(\boxed{i}) + U_\alpha(\boxed{i}) + V_\alpha(\boxed{i})$ *is even for all* $\boxed{i} \in V(\alpha)$
- $\deg(\textcircled{u})$ *is even and* $\deg(\textcircled{u}) \ge 4$ *for all* $\textcircled{u} \in V(\alpha)$
- $|E(\alpha)| \le n^\tau$

Remark 4.30. *Note that the shapes in $\mathcal{L}$ can have isolated vertices in $U_\alpha \cap V_\alpha$.*

Remark 4.31. *$\mathcal{L}$ captures all the shapes that have nonzero coefficient when we write $\mathcal{M}$ in terms of graph matrices. The constraint* $\deg(\textcircled{u}) \ge 4$ *arises because pseudocalibration gives us that* $\deg(\textcircled{u})$ *is even,* $\textcircled{u}$ *cannot be isolated, and $h_2(1) = 0$.*

For a shape α, we define

$$\alpha! := \prod_{e \in E(\alpha)} l(e)!$$

Note that this equals the factorial of the corresponding index of the Hermite polynomial for this shape.

Definition 4.32. *For any shape α, if $\alpha \in \mathcal{L}$, define*

$$\lambda_\alpha := \left(\prod_{\textcircled{u} \in V(\alpha)} h_{\deg(\textcircled{u})}(1)\right) \cdot \frac{1}{n^{(|U_\alpha|+|V_\alpha|+|E(\alpha)|)/2}} \cdot \frac{1}{\alpha!}$$

Otherwise, define $\lambda_\alpha := 0$.

Corollary 4.33. *Modulo the footnote*[3], $\mathcal{M} = \sum_{\text{shapes } \alpha} \lambda_\alpha M_\alpha$.

4.3.2 Boolean setting pseudocalibration

We now present the pseudocalibration for the Boolean setting. For the sequel, we need notation for vectors on a slice of the Boolean cube.

Definition 4.34 (Slice). *Let* $v \in \{\pm 1\}^n$ *and* $\theta \in \mathbb{Z}$. *The slice* $\mathcal{S}_v(\theta)$ *is defined as*

$$\mathcal{S}_v(\theta) := \left\{ d \in \{\pm 1\}^n \mid \langle v, d \rangle = \theta \right\}.$$

We use $\mathcal{S}_v(\pm\theta)$ *to denote* $\mathcal{S}_v(\theta) \cup \mathcal{S}_v(-\theta)$ *and* $\mathcal{S}(\theta)$ *to denote* $\mathcal{S}_v(\theta)$ *when* v *is the all-ones vector.*

Remark 4.35. *With our notation for the slice, the planted distribution in the Boolean setting can be equivalently described as*

1. *Sample* $v \in \left\{ \frac{\pm 1}{\sqrt{n}} \right\}^n$ *uniformly, and then*

2. *Sample* $d_1, \ldots, d_m$ *independently and uniformly from* $\mathcal{S}_{\sqrt{n} \cdot v}(\pm\sqrt{n})$.

The planted distribution doesn't actually exist for every n, but this is immaterial, as we can still define the pseudoexpectation via the same formula.

We will also need the expectation of monomials over the slice $\mathcal{S}(\sqrt{n})$ since they will appear in the description of the pseudocalibrated Fourier coefficients.

Definition 4.36. $e(k) := \mathbb{E}_{x \in_{\mathrm{R}} \mathcal{S}(\sqrt{n})} \left[x_1 \cdots x_k \right]$.

We now compute the Fourier coefficients of $\widetilde{\mathbb{E}}\, v^\beta$, where $\beta \in \mathbb{F}_2^n$. The Fourier basis when $d_1, \ldots, d_m \in_{\mathrm{R}} \{\pm 1\}^n$ is the set of parity functions. Thus a character can be specified by $\alpha \in (\mathbb{F}_2^n)^m$, where α is composed of m vectors $\alpha_1, \ldots, \alpha_m \in \mathbb{F}_2^n$. More precisely, the character χ_α associated to α is defined as

$$\chi_\alpha(d_1, \ldots, d_m) := \prod_{u=1}^{m} d_u^{\alpha_u}$$

We denote by $|\alpha|$ the number of non-zero entries of α and define $|\alpha_u|$ similarly. Thinking of α as an $m \times n$ matrix with entries in $\mathbb{F}_2$, we also define $\alpha^\intercal \in (\mathbb{F}_2^n)^m$.

[3] Technically, the graph matrices M_α have rows and columns indexed by all subsets of $\mathcal{C}_m \cup \mathcal{S}_n$. The submatrix with rows and columns from $\binom{\mathcal{S}_n}{\leq D/2}$ equals the candidate moment matrix for $\widetilde{\mathbb{E}}$.

Lemma 4.37. *We have*

$$\widetilde{\mathbb{E}}\, v^{\beta} = \frac{1}{n^{|\beta|/2}} \sum_{\substack{\alpha:\ |\alpha| \le n^{\tau}, \\ |\alpha_u| \text{ even}, \\ |\alpha_i^{\intercal}| \equiv \beta_i \pmod 2}} \prod_{u=1}^{m} e(|\alpha_u|) \cdot \chi_{\alpha_u}(d_u).$$

The set of nonzero coefficients has a similar structure as in the Gaussian case: the rows of α must have an even number of entries, and the i-th column must have parity matching β_i.

Proof. Given $\alpha \in (\mathbb{F}_2^n)^m$ with $|\alpha| \le n^{\tau}$, the pseudocalibration equation enforces by construction that

$$\mathop{\mathbb{E}}_{d_1,\dots,d_m \in \{\pm 1\}^n} (\widetilde{\mathbb{E}}\, v^{\beta})(d_1,\dots,d_m) \cdot \chi_{\alpha}(d_1,\dots,d_m) = \mathop{\mathbb{E}}_{\mathrm{pl}} v^{\beta} \cdot \chi_{\alpha}(d_1,\dots,d_m).$$

Computing the RHS above yields

$$\begin{aligned}
\mathop{\mathbb{E}}_{v \in \{\pm 1\}^n} \mathop{\mathbb{E}}_{d_1,\dots,d_m \in_{\mathrm{R}} \mathcal{S}_v(\pm\sqrt{n})} \left[v^{\beta} \prod_{u=1}^{m} \chi_{\alpha_u}(d_u) \right] &= \mathop{\mathbb{E}}_{v \in \{\pm 1\}^n} \mathop{\mathbb{E}}_{d_1,\dots,d_m \in_{\mathrm{R}} \mathcal{S}(\pm\sqrt{n})} \left[v^{\beta} \prod_{u=1}^{m} \chi_{\alpha_u}(v) \chi_{\alpha_u}(d_u) \right] \\
&= \mathop{\mathbb{E}}_{v \in \{\pm 1\}^n} \chi_{\alpha_1+\dots+\alpha_m+\beta}(v) \mathop{\mathbb{E}}_{d_1,\dots,d_m \in \mathcal{S}(\pm\sqrt{n})} \left[\prod_{i=1}^{m} \chi_{\alpha_i}(d_i) \right] \\
&= \mathbb{1}_{[\alpha_1+\dots+\alpha_m=\beta]} \cdot \prod_{i=1}^{m} \mathop{\mathbb{E}}_{d_i \in \mathcal{S}(\pm\sqrt{n})} [\chi_{\alpha_i}(d_i)] \\
&= \mathbb{1}_{[\alpha_1+\dots+\alpha_m=\beta]} \cdot \prod_{i=1}^{m} \mathbb{1}_{[|\alpha_i| \equiv 0 \pmod 2]} \cdot \prod_{i=1}^{m} e(|\alpha_i|).
\end{aligned}$$

Since we have a general expression for the Fourier coefficient of each character, applying Fourier inversion concludes the proof. □

Claim 4.38. $e(2) = 0$.

Proof. Fix $y \in \mathcal{S}(\sqrt{n})$. Note that $(\sum_{i=1}^{n} y_i)^2 = n$ implying $\sum_{i<j} y_i y_j = 0$. Using this fact, we get

$$\mathop{\mathbb{E}}_{x \in \mathcal{S}(\sqrt{n})} [x_1 x_2] = \mathop{\mathbb{E}}_{\sigma \in S_n} y_{\sigma(1)} y_{\sigma(2)} = 0,$$

concluding the proof. □

We can now express the moment matrix in terms of graph matrices.

Definition 4.39. *Let $\mathcal{L}_{bool}$ be the set of shapes in $\mathcal{L}$ from Definition 4.29 in which the edge labels are all 1.*

Remark 4.40. *$\mathcal{L}_{bool}$ captures all the shapes that have nonzero coefficient when we write $\mathcal{M}$ in terms of graph matrices. Similar to Remark 4.31, since $e(2) = 0$, we have the same condition $deg(\textcircled{u}) \ge 4$ for shapes in $\mathcal{L}_{bool}$.*

Definition 4.41. *For all shapes α, if $\alpha \in \mathcal{L}_{bool}$ define*

$$\lambda_\alpha := \frac{1}{n^{(|U_\alpha|+|V_\alpha|)/2}} \prod_{\textcircled{u} \in V(\alpha)} e(\deg(\textcircled{u}))$$

Otherwise, let $\lambda_\alpha := 0$.

Corollary 4.42. $\mathcal{M} = \sum\limits_{\textit{shapes } \alpha} \lambda_\alpha M_\alpha$

4.3.3 Unifying the analysis

It turns out that the analysis of the Boolean setting mostly follows from the analysis in the Gaussian setting. Initially, the Boolean pseudocalibration is essentially equal to the Gaussian pseudocalibration in which we have removed all shapes containing at least one edge with a label $k \geq 2$. The coefficients on the graph matrices will actually be slightly different, but they both admit an upper bound that is sufficient for our purposes (see Proposition 4.45 for the precise statement).

During the course of the analysis, we may multiply two graph matrices and produce graph matrices with improper parallel edges (the intersections terms). We will always bound the intersection terms by applying the triangle inequality and norm bounds. Since we show bounds that include the additional nonzero terms in the Gaussian case, the same bounds apply in the Boolean case.

We will consider separate cases at any point where the analysis differs between the two settings.

Furthermore, the lower bound should be *universal* for vectors d_i whose entries are iid from a distribution $\mathcal{D}$, provided that $\mathcal{D}$ is mean 0, variance 1, and $\mathcal{D}$ is $O(1)$-subgaussian (this doesn't cover some settings of interest where the d_i are not independent). The same proof should go through using graph matrices consisting of the orthogonal polynomials for $\mathcal{D}$, and the generalized norm bounds from [AMP20]. However, we have not checked all the details.

4.4 Proving PSD-ness

Here we give the details of the PSD-ness argument to prove Theorem 4.4. Fix a constant $\varepsilon > 0$ and a random instance $d_1, \ldots, d_m$ with $n \leq m \leq n^{3/2-\varepsilon}$. The candidate moment matrix $\mathcal{M}$ constructed in Section 4.3 goes up to SoS degree n^δ for a parameter $\delta > 0$.

For the pseudocalibration we truncate to only Fourier coefficients of size at most n^τ. The relationship between the parameters is $\delta \leq c\tau \leq c'\varepsilon$ where $c' < c < 1$ are absolute constants. We will assume that they are sufficiently small for all our proofs to go through. It is possible that truncation degree $O(n^\delta \log n)$ or $O(n^\delta/\varepsilon)$ instead of n^τ would suffice for our proof, but we have not attempted to optimize.

4.4.1 Handling non-spiders

Looking at the shapes that make up $\mathcal{M}$, the trivial shape with k square vertices contributes an identity matrix on the degree-$2k$ submatrix of $\mathcal{M}$. Our ultimate goal will be to bound all shapes against these identity matrices.

Recall the blocks of the moment matrix.

Definition 4.43 (Block). *For $k, l \in \{0, 1, \ldots, D/2\}$, the (k, l) block of $\mathcal{M}$ is the submatrix with rows from $\binom{[n]}{k}$ and columns from $\binom{[n]}{l}$. Note that when $\mathcal{M}$ is expressed as a sum of graph matrices, this exactly restricts $\mathcal{M}$ to shapes α with $|U_\alpha| = k$ and $|V_\alpha| = l$.*

We define the parameter $\eta := 1/\sqrt{n}$. The trivial shapes live in the diagonal blocks of $\mathcal{M}$, and on the (k, k) block contribute a factor of $\frac{1}{n^k} = \eta^{2k}$ on the diagonal. In principle, we could make η as small as we like[4] by considering the moments of a rescaling of v rather than v itself. Counter-intuitively, it will turn out that the scaling helps us prove PSD-ness (see [GJJ+20, Appendix D] for more details). It turns out that pseudocalibrating v as a unit vector (equivalently, using $\eta = 1/\sqrt{n}$) is sufficient for our analysis.

Towards the goal of bounding $\mathcal{M}$ by the identity terms, we will bound the norm of matrices on each block of $\mathcal{M}$, and invoke the following lemma to conclude PSD-ness.

Proposition 4.44. *Suppose a symmetric matrix $\mathcal{A} \in \mathbb{R}^{\binom{[n]}{\leq D} \times \binom{[n]}{\leq D}}$ satisfies, for some parameter $\eta \in (0, 1)$,*

1. *For each $k \in \{0, 1, \ldots, D\}$, the (k, k) block has minimum singular value at least $\eta^{2k}(1 - \frac{1}{D+1})$*
2. *For each $k, l \in \{0, 1, \ldots, D\}$ such that $k \neq l$, the (k, l) block has norm at most $\frac{\eta^{k+l}}{D+1}$.*

Then $\mathcal{A} \succeq 0$.

Proof. We need to show that for all vectors x, $x^\intercal \mathcal{A} x \geq 0$. Given a vector x, let $x_0, \ldots, x_D$ be its components in blocks $0, \ldots, D$. Observe that

$$x^\intercal \mathcal{A} x \geq \sum_{k \in [0,D]} \eta^{2k} \left(1 - \frac{1}{D+1}\right) \|x_k\|^2 - \sum_{k \neq l \in [0,D]} \frac{\eta^{k+l}}{D+1} \|x_k\| \|x_l\|$$

$$= (\|x_0\|, \eta\|x_1\|, \ldots, \eta^D\|x_D\|) \begin{pmatrix} 1 - \frac{1}{D+1} & -\frac{1}{D+1} & \cdots & -\frac{1}{D+1} \\ -\frac{1}{D+1} & 1 - \frac{1}{D+1} & \cdots & -\frac{1}{D+1} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{D+1} & -\frac{1}{D+1} & \cdots & 1 - \frac{1}{D+1} \end{pmatrix} \begin{pmatrix} \|x_0\| \\ \eta\|x_1\| \\ \vdots \\ \eta^D\|x_D\| \end{pmatrix} \geq 0.$$

□

[4] Though pseudocalibration truncation errors may become non-negligible for extremely tiny η.

We have the following general-purpose upper bound on the coefficients of the graph matrices.

Proposition 4.45. $|\lambda_\alpha| \leq \eta^{|U_\alpha|+|V_\alpha|} \cdot \frac{|E(\alpha)|^{3\cdot|E(\alpha)|}}{n^{|E(\alpha)|/2}}$

Proof. **(Gaussian setting)** Recall that the coefficients λ_α are either zero or are defined by the formula

$$\lambda_\alpha = \eta^{|U_\alpha|+|V_\alpha|} \cdot \left(\prod_{\textcircled{u} \in V(\alpha)} h_{\deg(\textcircled{u})}(1) \right) \cdot \frac{1}{n^{|E(\alpha)|/2}} \cdot \frac{1}{\alpha!}$$

The sequence $h_k(1)$ satisfies the recurrence $h_0(1) = h_1(1) = 1, h_{k+1}(1) = h_k(1) - kh_{k-1}(1)$. We can prove by induction that $|h_k(1)| \leq k^k$ and hence,

$$\prod_{\textcircled{u} \in V(\alpha)} \left| h_{\deg(\textcircled{u})}(1) \right| \leq \prod_{\textcircled{u} \in V(\alpha)} (\deg(\textcircled{u}))^{\deg(\textcircled{u})} \leq |E(\alpha)|^{|E(\alpha)|}.$$

(Boolean setting) In the boolean setting the coefficients λ_α are defined by

$$\lambda_\alpha = \eta^{|U_\alpha|+|V_\alpha|} \cdot \left(\prod_{\textcircled{u} \in V(\alpha)} e(\deg(\textcircled{u})) \right)$$

Using [GJJ+20, Corollary B.12], we have that $|e(k)| \leq k^{3k} \cdot n^{-k/2}$. Thus,

$$|\lambda_\alpha| = \eta^{|U_\alpha|+|V_\alpha|} \cdot \prod_{\textcircled{u} \in V(\alpha)} |e(\deg(\textcircled{u}))| \leq \eta^{|U_\alpha|+|V_\alpha|} \cdot \frac{|E(\alpha)|^{3|E(\alpha)|}}{n^{|E(\alpha)|/2}}.$$

□

The specific norm bounds we use are the following. See [GJJ+20, Appendix A] for the proofs.

Lemma 4.46 (Boolean setting norm bound)**.** *There is a universal constant C such that the following norm bound holds for all proper shapes α, allowing isolated vertices, whp:*

$$\|M_\alpha\| \leq 2 \cdot (|V(\alpha)| \cdot \log(n))^{C \cdot |V(\alpha) \setminus (U_\alpha \cap V_\alpha)|} \cdot n^{\frac{w(V(\alpha)) - w(S_{\min}) + w(I_\alpha)}{2}}$$

For a Hermite shape α, define the *total size* to be $|U_\alpha| + |V_\alpha| + |W_\alpha| + |E(\alpha)|$.

Lemma 4.47 (Gaussian setting norm bound)**.** *There is a universal constant C such that the following norm bound holds for all proper shapes α, allowing isolated vertices, with total size at most n whp:*

$$\|M_\alpha\| \leq 2 \cdot (|V(\alpha)| \cdot (1 + |E(\alpha)|) \cdot \log(n))^{C \cdot (|V(\alpha) \setminus (U_\alpha \cap V_\alpha)| + |E(\alpha)|)} \cdot n^{\frac{w(V(\alpha)) - w(S_{\min}) + w(I_\alpha)}{2}}$$

Lemma 4.48. *If $\alpha \in \mathcal{L}$ is not a trivial shape and not a spider, then*

$$|\lambda_\alpha| \|M_\alpha\| \le \frac{\eta^{|U_\alpha|+|V_\alpha|}}{n^{\Omega(\varepsilon|E(\alpha)|)}} .$$

Proof. This is the formal statement of Lemma 4.17. Using the formal norm bounds in the proof, the only factor we have not handled is the log factor in the norm bound,

$$2 \cdot (|V(\alpha)| \cdot (1 + |E(\alpha)|) \cdot \log(n))^{C\cdot(|V(\alpha)\setminus(U_\alpha \cap V_\alpha)|+|E(\alpha)|)} .$$

Observe that since there no degree 0 vertices in $V(\alpha) \setminus (U_\alpha \cap V_\alpha)$, we have that $|V(\alpha) \setminus (U_\alpha \cap V_\alpha)| \le 2|E(\alpha)|$. We also have $|V(\alpha)| \cdot (1 + |E(\alpha)|) \cdot \log n \le n^{O(\tau)}$. The log factor is upper bounded by:

$$2 \cdot (|V(\alpha)| \cdot (1 + |E(\alpha)|) \cdot \log(n))^{C\cdot(|V(\alpha)\setminus(U_\alpha \cap V_\alpha)|+|E(\alpha)|)} \le n^{O(\tau|E(\alpha)|)} .$$

Therefore, it can be absorbed into $\frac{1}{n^{\Omega(\varepsilon|E(\alpha)|)}}$ without a qualitative change. □

This says that nontrivial non-spider shapes have $o_n(1)$ norm (ignoring the extra factor η for the moment). We now demonstrate how to use this norm bound to control the total norm of all non-spiders in a block of $\mathcal{M}$, Corollary 4.50. We will first need a couple propositions which will also be of use to us later after we kill the spiders.

Proposition 4.49. *The number of proper shapes with at most L vertices and exactly k edges is at most $L^{8(k+1)}$.*

Proof. The following process captures all shapes (though many will be constructed multiple times):

- Choose the number of square and circle variables in each of the four sets $U \cap V, U \setminus (U \cap V), V \setminus (U \cap V), W$. This contributes a factor of L^8.
- Place each edge between two of the vertices. This contributes a factor of L^{2k}.

□

Corollary 4.50. *For $k, l \in \{0, 1, \dots, D/2\}$, let $\mathcal{B}_{k,l} \subseteq \mathcal{L}$ denote the set of nontrivial, non-spiders $\alpha \in \mathcal{L}$ on the (k, l) block i.e. $|U_\alpha| = k, |V_\alpha| = l$. The total norm of the non-spiders in $\mathcal{B}_{k,l}$ satisfies*

$$\sum_{\alpha \in \mathcal{B}_{k,l}} |\lambda_\alpha| \|M_\alpha\| \le \eta^{k+l} \cdot \frac{1}{n^{\Omega(\varepsilon)}}$$

Proof.

$$\begin{aligned}
\sum_{\alpha \in \mathcal{B}_{k,l}} |\lambda_\alpha| \|M_\alpha\| &\leq \eta^{k+l} \cdot \sum_{\alpha \in \mathcal{B}_{k,l}} \frac{1}{n^{\Omega(\varepsilon |E(\alpha)|)}} && \text{(Lemma 4.48)} \\
&\leq \eta^{k+l} \cdot \sum_{i=1}^{\infty} \frac{n^{O(\tau i)}}{n^{\Omega(\varepsilon i)}} && \text{(Proposition 4.49 and } |E(\alpha)| \geq 1 \text{ for } \alpha \in \mathcal{B}_{k,l}) \\
&= \eta^{k+l} \cdot \frac{1}{n^{\Omega(\varepsilon)}}
\end{aligned}$$

□

4.4.2 Killing a single spider

We saw in the Proof Outline that the shape $2\beta_1 + \frac{1}{n}\beta_2$ lies in the nullspace of a moment matrix which satisfies the constraints "$\langle v, d_u \rangle^2 = 1$". The shape β_1 is exactly the kind of substructure that appears in a spider! Therefore it is natural to hope that if α is a left spider, then $\mathcal{M}M_\alpha = 0$. This doesn't quite hold because $\langle v, d_u \rangle^2$ is "missing" some intersection terms: in realizations of α, the end vertices are required to be distinct from the other squares in α, which prevents terms for all pairs i, j from appearing in the product $\mathcal{M}M_\alpha$. There are smaller "intersection terms" (which we call collapses of α here) that we can add so that the end vertices are permitted to take on all pairs i, j. After adding in these terms, we will produce a matrix L with $\mathcal{M}L = 0$.

We first define what it means to collapse a shape into another shape by merging two vertices. Here, we only define it for merging two square vertices, since these are the only kind of merges that will happen in our analysis of intersection terms.

Definition 4.51 (Improper collapse)**.** *Let α be a shape and let $\boxed{i}, \boxed{j}$ be two distinct square vertices in $V(\alpha)$. We define the improper collapse of $\boxed{i}, \boxed{j}$ by:*

- *Remove $\boxed{i}, \boxed{j}$ from $V(\alpha)$ and replace them by a single new vertex $\boxed{k}$.*
- *Replace each edge $\{\boxed{i}, \textcircled{u}\}$ and $\{\boxed{j}, \textcircled{u}\}$, if present, by $\{\boxed{k}, \textcircled{u}\}$, keeping the same labels (note that there may be multiedges and so the new shape may not be proper).*
- *Set $U(\boxed{k}) = U(\boxed{i}) + U(\boxed{j}) \pmod 2$ and $V(\boxed{k}) = V(\boxed{i}) + V(\boxed{j}) \pmod 2$.*

Definition 4.52 (Collapsing a shape)**.** *Let α be a shape with two distinct square vertices $\boxed{i}, \boxed{j}$. We say that β is a (proper) collapse of $\boxed{i}, \boxed{j}$ if β has nonzero coefficient in the linearization of the improper collapse of $\boxed{i}, \boxed{j}$.*

Remark 4.53. *If $l_1, \ldots, l_k$ are the labels of a set of parallel edges, then the product $h_{l_1}(z) \cdots h_{l_k}(z)$ is even/odd depending on the parity of $l_1 + \cdots + l_k$. Thus the nonzero Fourier coefficients in the linearization will be the terms of matching parity. Therefore, in both the Boolean and Gaussian cases, the shapes that are proper collapses of a given improper collapse are formed by replacing each* *set of parallel edges by a single edge e such that* $l(e) \leq l_1 + \ldots + l_k$ *and* $l(e) \equiv l_1 + \cdots + l_k \pmod 2$.

Remark 4.54. *Looking at the definition and in light of the previous remark, we have the following.*

1. *The number of circle vertices does not change by collapsing a shape but the number of square vertices decreases by* 1.

2. $\alpha \in \mathcal{L}$ *has the property that the vertices have odd degree if and only if they are in* $(U_\alpha \cup V_\alpha) \setminus (U_\alpha \cap V_\alpha)$. *When* α *collapses, this property is preserved.*

We now define the desired shapes L_k which lie in the null space of $\mathcal{M}$.

Definition 4.55. *For* $k \geq 2$ *define the shape* ℓ_k *on* $\{\boxed{1}, \ldots, \boxed{k}, \textcircled{1}\}$ *with two edges* $\{\{\boxed{1}, \textcircled{1}\}, \{\boxed{2}, \textcircled{1}\}\}$. *The left side of* ℓ_k *consists of* $U_{\ell_k} = \{\boxed{1}, \ldots, \boxed{k}\}$. *The right side consists of* $V_{\ell_k} = \{\boxed{3}, \ldots, \boxed{k}, \textcircled{1}\}$.

Definition 4.56. *Define the "completed" version* L_k *of* ℓ_k *to be the matrix which is the sum of* $c_\beta M_\beta$ *for* β *being the following shapes with coefficients:*

- *(*$L_{k,1}$*):* ℓ_k, *with coefficient 2.*

- *(*$L_{k,2}$*): If* $k \geq 3$, *collapse* $\boxed{1}$ *and* $\boxed{3}$ *in* ℓ_k *with coefficient* $\frac{2}{n}$

- *(*$L_{k,3}$*): If* $k \geq 4$, *collapse* $\boxed{1}$ *and* $\boxed{3}$, *and collapse* $\boxed{2}$ *and* $\boxed{4}$ *in* ℓ_k *with coefficient* $\frac{2}{n^2}$

- *(*$L_{k,4}$*): Collapse* $\boxed{1}$ *and* $\boxed{2}$, *replacing the edges by an edge with label 2, with coefficient* $\frac{1}{n}$

- *(*$L_{k,5}$*): If* $k \geq 3$, *collapse* $\boxed{1}, \boxed{2}$, *and* $\boxed{3}$, *replacing the edges by an edge with label 2, with coefficient* $\frac{1}{n}$.

For a pictorial representation of the ribbons/shapes, see Fig. 4.8 below.

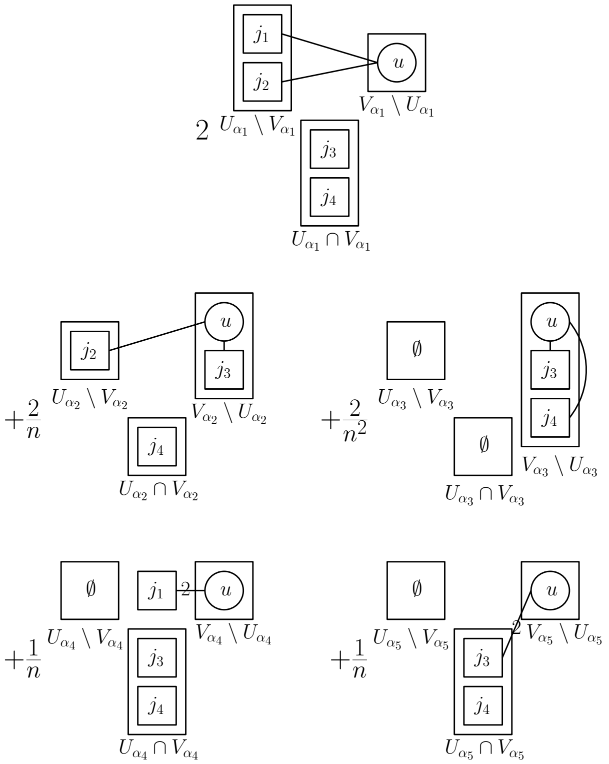
Lemma 4.57. *If* $\mathcal{M}$ *is a moment matrix satisfying the constraints "*$\langle v, d_u \rangle^2 = 1$*", then* $\mathcal{M} L_k = 0$

Proof. These shapes are constructed so that if we fix a partial realization of the vertices $\textcircled{1}$ and $\boxed{3}, \ldots, \boxed{k}$ as $\textcircled{u} \in \mathcal{C}_m$ and $S \in \binom{\mathcal{S}_n}{k-2}$, the squares $\boxed{1}$ and $\boxed{2}$ can still be realized as any $j_1, j_2 \in [n]$. That is, exactly the following equality holds,

$$\begin{aligned}
(\mathcal{M} L_k)_I &= \sum_{\substack{\textcircled{u} \in \mathcal{C}_m, \\ S \in \binom{\mathcal{S}_n}{k-2}}} \left(\sum_{\substack{j_1, j_2 \in [n]: \\ j_1 \neq j_2}} \widetilde{\mathbb{E}}[v^I v^S v_{j_1} v_{j_2}] d_{uj_1} d_{uj_2} + \sum_{j_1 \in [n]} \widetilde{\mathbb{E}}[v^I v^S v_{j_1}^2](d_{uj_1}^2 - 1) \right) \\
&= \sum_{\substack{\textcircled{u} \in \mathcal{C}_m, \\ S \in \binom{\mathcal{S}_n}{k-2}}} \widetilde{\mathbb{E}}[v^I v^S (\langle v, d_u \rangle^2 - 1)] \\
&= 0
\end{aligned}$$

To demonstrate how the coefficients arise, we analyze the ribbons R which L_k is composed of and see how they contribute to the output. For pictures of the ribbons/shapes, see Fig. 4.8 below. Let the ribbon be partially realized as $\textcircled{u}$ and $S = \{\boxed{j_3}, \ldots, \boxed{j_k}\}$. Let $(ML_k)_{I(u,S)}$ denote the terms in $(ML_k)_I$ with this partial realization. In this notation we want to show

$$(\mathcal{M}L_k)_{I(u,S)} = \sum_{\substack{j_1, j_2 \in [n]: \\ j_1 \neq j_2}} \widetilde{\mathbb{E}}[v^I v^S v_{j_1} v_{j_2}] d_{uj_1} d_{uj_2} + \sum_{j_1 \in [n]} \widetilde{\mathbb{E}}[v^I v^S v_{j_1}^2](d_{uj_1}^2 - 1).$$



j_1
j_2
u
$V_{\alpha_1} \setminus U_{\alpha_1}$
2
$U_{\alpha_1} \setminus V_{\alpha_1}$
j_3
j_4
$U_{\alpha_1} \cap V_{\alpha_1}$
u
j_2
j_3
$+\frac{2}{n}$
$U_{\alpha_2} \setminus V_{\alpha_2}$
$V_{\alpha_2} \setminus U_{\alpha_2}$
j_4
$U_{\alpha_2} \cap V_{\alpha_2}$
$\emptyset$
u
j_3
j_4
$U_{\alpha_3} \setminus V_{\alpha_3}$
$+\frac{2}{n^2}$
$\emptyset$
$V_{\alpha_3} \setminus U_{\alpha_3}$
$U_{\alpha_3} \cap V_{\alpha_3}$
$\emptyset$
j_1
2
u
$+\frac{1}{n}$
$U_{\alpha_4} \setminus V_{\alpha_4}$
$V_{\alpha_4} \setminus U_{\alpha_4}$
j_3
j_4
$U_{\alpha_4} \cap V_{\alpha_4}$
$\emptyset$
u
$+\frac{1}{n}$
$U_{\alpha_5} \setminus V_{\alpha_5}$
2
$V_{\alpha_5} \setminus U_{\alpha_5}$
j_3
j_4
$U_{\alpha_5} \cap V_{\alpha_5}$

Figure 4.8: The five shapes that make up L_4.

1. If we take a ribbon R with $A_R = \{\boxed{j_1}, \ldots, \boxed{j_k}\}$, $B_R = \{\boxed{j_3}, \ldots, \boxed{j_k}\} \cup \{\textcircled{u}\}$ and $E(R) = \{\{\boxed{j_1}, \textcircled{u}\}, \{\boxed{j_2}, \textcircled{u}\}\}$ where $j_1 \neq j_2$ and $j_1, j_2 \notin S$ then

   $$(\mathcal{M}M_R)_{I(u,S)} = \widetilde{\mathbb{E}}[v^I v^S v_{j_1} v_{j_2}] d_{uj_1} d_{uj_2}.$$

   This ribbon must "cover" both ordered pairs (j_1, j_2) and (j_2, j_1), so we want each such ribbon R to appear with a coefficient of 2 in L_k.

2. If we take a ribbon R with $A_R = \{\boxed{j_1}, \ldots, \boxed{j_k}\} \setminus \{\boxed{j_1}, \boxed{j_3}\}$, $B_R = \{\boxed{j_3}, \ldots, \boxed{j_k}\} \cup \{\textcircled{u}\}$

and $E(R) = \{\{\boxed{j_3}, \textcircled{u}\}, \{\boxed{j_2}, \textcircled{u}\}\}$ where $j_1 = j_3 \in S$ then

$$(\mathcal{M}M_R)_{I(u,S)} = \widetilde{\mathbb{E}}[v^I v^{S \setminus \{j_3\}} v_{j_2}] d_{uj_3} d_{uj_2} = n\, \widetilde{\mathbb{E}}[v^I v^S v_{j_1} v_{j_2}] d_{uj_1} d_{uj_2}.$$

Taking a coefficient of $\frac{2}{n}$ in L_k covers the two pairs (j_1, j_2) and (j_2, j_1) for this case of overlap with S.

3. If we take a ribbon R with $A_R = \{\boxed{j_1}, \ldots, \boxed{j_k}\} \setminus \{\boxed{j_1}, \boxed{j_2}, \boxed{j_3}, \boxed{j_4}\}$, $B_R = \{\boxed{j_3}, \ldots, \boxed{j_k}\} \cup \{\textcircled{u}\}$ and $E(R) = \{\{\boxed{j_3}, \textcircled{u}\}, \{\boxed{j_4}, \textcircled{u}\}\}$ where $j_1 = j_3 \in S$ and $j_2 = j_4 \in S$ then

$$(\mathcal{M}M_R)_{I(u,S)} = \widetilde{\mathbb{E}}[v^I v^{S \setminus \{j_3, j_4\}}] d_{uj_3} d_{uj_4} = n^2\, \widetilde{\mathbb{E}}[v^I v^S v_{j_1} v_{j_2}] d_{uj_1} d_{uj_2}.$$

Taking a coefficient of $\frac{2}{n^2}$ in L_k covers the two pairs (j_1, j_2) and (j_2, j_1) for this case of overlap with S.

4. If we take a ribbon R with $A_R = \{\boxed{j_1}, \ldots, \boxed{j_k}\} \setminus \{\boxed{j_1}, \boxed{j_2}\}$, $B_R = \{\boxed{j_3}, \ldots, \boxed{j_k}\} \cup \{\textcircled{u}\}$ and $E(R) = \{\{\boxed{j_1}, \textcircled{u}\}_2\}$ where $j_1 = j_2 \notin S$ then

$$(\mathcal{M}M_R)_{I(u,S)} = \widetilde{\mathbb{E}}[v^I v^S](d_{uj_1}^2 - 1) = n\, \widetilde{\mathbb{E}}[v^I v^S v_{j_1}^2](d_{uj_1}^2 - 1).$$

Taking a coefficient of $\frac{1}{n}$ in L_k covers these terms.

5. If we take a ribbon R with $A_R = \{\boxed{j_1}, \ldots, \boxed{j_k}\} \setminus \{\boxed{j_1}, \boxed{j_2}\}$, $B_R = \{\boxed{j_3}, \ldots, \boxed{j_k}\} \cup \{\textcircled{u}\}$ and $E(R) = \{\{\boxed{j_3}, \textcircled{u}\}_2\}$ where $j_1 = j_2 = j_3 \in S$ then

$$(\mathcal{M}M_R)_{I(u,S)} = \widetilde{\mathbb{E}}[v^I v^S](d_{uj_3}^2 - 1) = n\, \widetilde{\mathbb{E}}[v^I v^S v_{j_1}^2](d_{uj_1}^2 - 1).$$

Taking a coefficient of $\frac{1}{n}$ in L_k covers these terms.

□

Proposition 4.58. *Let α, β be composable shapes. Assume that $V(\alpha) \setminus V_\alpha$ has only square vertices. Let $\widetilde{\mathcal{I}} = \{\alpha_P : P \in \mathcal{P}_{\alpha,\beta}\}$ be the set of improper collapses (intersection terms) in $M_\alpha M_\beta$. Then there are coefficients c_γ for $\gamma \in \widetilde{\mathcal{I}}$, $|c_\gamma| \le 2^{|V(\alpha) \setminus V_\alpha|} |V(\gamma)|^{|V)\alpha) \setminus U_\alpha|}$, such that*

$$M_\alpha \cdot M_\beta = \sum_{\gamma \in \widetilde{\mathcal{I}}} c_\gamma M_\gamma\,.$$

Proof. The coefficients exist by Proposition 2.39. We need to upper bound them by upper bounding the coefficient of a ribbon from the right-hand side.

Let T be a ribbon on the right-hand side of shape γ. We say ribbons R of shape α and S of shape β contribute to T if $M_R M_S = M_T$. From T, we can completely recover the sets A_R and B_S. The labels of $V(R) \setminus A_R$ must be among the labels of T; choose them in at most $|V(\gamma)|^{|V(\alpha) \setminus U_\alpha|}$ ways. This also determines $B_R = A_S$. All that remains is to determine the graph structure of S. Since improper collapsing doesn't lose any edges, knowing the labels of R we know exactly which edges of T must come from R and S. The vertices $V(T) \setminus V(R)$ must come from S, as must B_R; pick a subset of $V(R) \setminus B_R$ to include in $2^{|V(\alpha) \setminus V_\alpha|}$ ways. $\square$

Let α be a left spider with end vertices $\boxed{i}, \boxed{j}$ which are adjacent to a circle $\textcircled{u}$. Recall that our goal is to argue that $\mathcal{M}M_\alpha \approx 0$. To get there, we can try and factor M_α across the vertex separator $S = U_\alpha \cup \{\textcircled{u}\} \setminus \{\boxed{i}, \boxed{j}\}$ which separates α into

$$M_\alpha \approx L_{|U_\alpha|} \cdot M_{\text{body}(\alpha)}$$

where we have defined,

Definition 4.59. *Let α be a left spider with end vertices $\boxed{i}, \boxed{j}$. Define* $\text{body}(\alpha)$ *as the shape whose graph is α with $\boxed{i}$ and $\boxed{j}$ deleted and with $U_{\text{body}(\alpha)} = U_\alpha \cup \{\textcircled{u}\} \setminus \{\boxed{i}, \boxed{j}\}, V_{\text{body}(\alpha)} = V_\alpha$. The definition is analogous for right spiders.*

Due to Lemma 4.57, the right-hand side of the approximation is in the null space of $\mathcal{M}$. We now formalize this approximate factorization.

Definition 4.60. *Let α be a spider with end vertices $\boxed{i}, \boxed{j}$. Define $\widetilde{\mathcal{I}}_\alpha$ to be the set of shapes that can be obtained from α by performing at least one of the following steps:*

- *Improperly collapse $\boxed{i}$ with a square vertex in α*
- *Improperly collapse $\boxed{j}$ with a square vertex in α*

Let $\mathcal{I}_\alpha$ be the set of proper shapes that can be obtained via the same process but using proper collapses.

In the above definition, we allow $\boxed{i}, \boxed{j}$ to collapse with two distinct squares, or to collapse together, or to both collapse with a common third vertex. For technical reasons we need to work with a refinement of $\mathcal{I}_\alpha$ into two sets of shapes and use tighter bounds on coefficients of one set.

Definition 4.61. *Let $\mathcal{I}_\alpha^{(1)}$ be the set of shapes that can be obtained from α by performing at least one of the following steps:*

- *Collapse $\boxed{i}$ with a square vertex in* $\text{body}(\alpha) \setminus U_\alpha$
- *Collapse $\boxed{j}$ with a square vertex in* $\text{body}(\alpha) \setminus U_\alpha$ *(distinct from $\boxed{i}$'s collapse if it happened)*

Let $\mathcal{I}_\alpha^{(2)} := \mathcal{I}_\alpha \setminus \mathcal{I}_\alpha^{(1)}$ and define the improper versions $\widetilde{\mathcal{I}}_\alpha^{(1)}, \widetilde{\mathcal{I}}_\alpha^{(2)}$ analogously.

Lemma 4.62. *Let α be a left spider with end vertices $\boxed{i}, \boxed{j}$. There are coefficients c_β for $\beta \in \widetilde{\mathcal{I}}_\alpha$ such that*

$$L_{|U_\alpha|} \cdot M_{\text{body}(\alpha)} = 2M_\alpha + \sum_{\beta \in \widetilde{\mathcal{I}}_\alpha} c_\beta M_\beta,$$

$$\left|c_\beta\right| \le \begin{cases} 40|V(\alpha)|^3 & \beta \in \widetilde{\mathcal{I}}_\alpha^{(1)} \\ \frac{40|V(\alpha)|^3}{n} & \beta \in \widetilde{\mathcal{I}}_\alpha^{(2)} \end{cases}.$$

Proof. First, we can check that the coefficient of M_α is 2. Only the ℓ_k term of L_k has the full number of squares, and it has a factor of 2 in L_k.

The shapes in $\widetilde{\mathcal{I}}_\alpha$ are definitionally the intersection terms that appear in this graph matrix product, and furthermore the shapes in $\widetilde{\mathcal{I}}_\alpha$ are definitionally the intersection terms for the ℓ_k term. Using Proposition 4.58, for each of the five shapes in $L_{|U_\alpha|}$ the coefficient it contributes is bounded by $4|V(\alpha)|^3$. The coefficient on ℓ_k is 2, so the coefficients for $\widetilde{\mathcal{I}}_\alpha^{(1)}$ are at most $8|V(\alpha)|^3$. The maximum coefficient of the other four shapes in $L_{|U_\alpha|}$ is $\frac{2}{n}$, so their total contribution to coefficients on $\widetilde{\mathcal{I}}_\alpha^{(2)}$ is at most $\frac{32|V(\alpha)|^3}{n}$. □

Proposition 4.63. *Let $l_1 \le \cdots \le l_k \in \mathbb{N}$ and let $L = l_1 + \cdots + l_k$. Assume $L \ge 1$. In the Fourier expansion of $h_{l_1}(z)\cdots h_{l_k}(z)$, the maximum coefficient is bounded in magnitude by $(2L)^{L-l_k}$.*

Proof. In the boolean case, the coefficient is 1. In the Gaussian case, the "linearization coefficient" of $h_p(z)$ in this product is given by orthogonality to be

$$\frac{\mathbb{E}_{z\sim\mathcal{N}(0,1)}[h_{l_1}(z)\cdots h_{l_k}(z)\cdot h_p(z)]}{\mathbb{E}_{z\sim\mathcal{N}(0,1)}[h_p^2(z)]} = \frac{\mathbb{E}_{z\sim\mathcal{N}(0,1)}[h_{l_1}(z)\cdots h_{l_k}(z)\cdot h_p(z)]}{p!}$$

A formula from, e.g., [RW97, Example G (Continued)] shows that $\mathbb{E}[h_{l_1}\cdots h_{l_k}\cdot h_p]$ equals the number of "block perfect matchings": perfect matchings on $l_1 + \cdots + l_k + p$ elements divided into blocks of size l_i or p such that no two elements from the same block are matched. Bound the number of block perfect matchings by:

- Pick a partial function from blocks $l_1, \ldots, l_{k-1}$ to $[L]$ in at most $(L+1)^{L-l_k}$ ways.
- If this forms a valid partial matching and there are p unmatched elements remaining, match them with the elements from the block of size p in $p!$ ways.

Therefore the coefficient is bounded by $(L+1)^{L-l_k} \le (2L)^{L-l_k}$. □

Lemma 4.64. *If α is a left spider, there are coefficients c_β for each $\beta \in \mathcal{I}_\alpha$ such that*

$$L_{|U_\alpha|} \cdot M_{\text{body}(\alpha)} = 2M_\alpha + \sum_{\beta\in\mathcal{I}_\alpha} c_\beta M_\beta,$$

$$\left|c_\beta\right| \le \begin{cases} 160|V(\alpha)|^7|E(\alpha)|^2 & \beta \in \mathcal{I}_\alpha^{(1)} \\ \frac{160|V(\alpha)|^7|E(\alpha)|^2}{n} & \beta \in \mathcal{I}_\alpha^{(2)} \end{cases}.$$

Proof. We convert each $M_\beta, \beta \in \widetilde{\mathcal{I}}_\alpha$ in Lemma 4.62 into a sum of proper shapes.

$$\begin{aligned} M_\beta &= \sum_{\text{ribbons } R \text{ of shape } \beta} M_R \\ &= \frac{1}{|\operatorname{Aut}(\beta)|} \sum_{\substack{\text{injective, type-preserving} \\ \phi: V(\beta) \to \mathcal{C}_m \cup \mathcal{S}_n}} M_{\phi(\beta)} \qquad \text{(Fact 2.25)}\,. \end{aligned}$$

Each of the improper ribbons $M_{\phi(\beta)}$ is Fourier expanded as a sum of proper ribbons. Let $C_{Fourier}$ be an upper bound on the Fourier coefficients. Collecting the resulting ribbons by shape, let $c_{\beta'}$ be the Fourier coefficient.

$$\begin{aligned} M_\beta &= \frac{1}{|\operatorname{Aut}(\beta)|} \sum_{\beta'} c_{\beta'} \sum_{\substack{\text{injective, type-preserving} \\ \phi: V(\beta') \to \mathcal{C}_m \cup \mathcal{S}_n}} M_{\phi(\beta')} \\ &= \frac{|\operatorname{Aut}(\beta')|}{|\operatorname{Aut}(\beta)|} \sum_{\beta'} c_{\beta'} M_{\beta'}\,. \end{aligned}$$

Let C_{Aut} be an upper bound on the ratio $\frac{|\operatorname{Aut}(\beta')|}{|\operatorname{Aut}(\beta)|}$.

We use the following bounds on $C_{Fourier}$ and C_{Aut}. The only improperness in β comes from collapsing (at most) the two end vertices, which have a single incident edge each. Therefore the set of labels of any parallel edges is either $\{1, k\}$ or $\{1, 1, k\}$, for some $k \le |E(\alpha)|$. By Proposition 4.63, we have $C_{Fourier} \le 4|E(\alpha)|^2$. There are at most two extra parallel edges in β, so we have $C_{Aut} \le |V(\alpha)|^4$ using Proposition 2.26. Therefore the coefficients increase by at most $C_{Fourier} \cdot C_{Aut} \le 4|E(\alpha)|^2|V(\alpha)|^4$. □

Corollary 4.65. *If α is a right spider, there are coefficients c_β with the same bounds given in Lemma 4.64 such that*

$$M_{\operatorname{body}(\alpha)} \cdot L^{\intercal}_{|U_\alpha|} = 2M_\alpha + \sum_{\beta \in \mathcal{I}_\alpha} c_\beta M_\beta.$$

Corollary 4.66. *If $\mathcal{M}$ satisfies the constraints "$\langle v, d_u \rangle^2 = 1$", $x \perp \operatorname{Null}(\mathcal{M})$ and α is a spider, then for some c_β with the same bounds given in Lemma 4.64,*

$$x^\top (M_\alpha - \sum_{\beta \in \mathcal{I}_\alpha} c_\beta M_\beta) x = 0$$

Proof. For a left spider, since

$$\mathcal{M}(2M_\alpha + \sum_{\beta \in \mathcal{I}_\alpha} c_\beta M_\beta) = \mathcal{M} \cdot L_{|U_\alpha|} \cdot M_{\alpha'} = 0$$

every column of $2M_\alpha + \sum_{\beta \in \mathcal{I}_\alpha} c_\beta M_\beta$ is in $\mathrm{Null}(\mathcal{M})$ and the claim follows. For a right spider, the proof is analogous. □

4.4.3 Proving PSD-ness: killing all the spiders

We must bound the accumulation on the coefficients λ'_β. We do this by considering the *web* of spiders and non-spiders created by each spider and using bounds on the c_β and λ_α to argue that the contributions do not blow up, via an interesting charging scheme that exploits the structure of these graphs.

The strategy is to start with the moment matrix $\mathcal{M}$ and apply Corollary 4.66 repeatedly until we end up with no spiders in our decomposition. For each spider, killing it via Corollary 4.66 leaves only intersection terms. Some of those intersection terms may themselves be smaller spiders, in which case we will apply the corollary again and again until only non-spiders remain. The difficulty during this procedure is to bound the total coefficient accumulated on each non-spider. To capture this process, we define the web of a spider α, which will be a directed acyclic graph that will capture the spider killing process. For the sake of distinction, we will call the vertices of this graph "nodes".

Definition 4.67 (Web of α)**.** *The web $W(\alpha)$ of a spider α is a rooted directed acyclic graph (DAG) whose nodes are shapes and whose root is α. Each spider node γ has edges to nodes β for each shape $\beta \in \mathcal{I}_\gamma$. The non-spider nodes are leaves/sinks of the DAG.*

Remark 4.68. *The DAG structure arises because each shape in $\mathcal{I}_\gamma$ has strictly fewer square vertices than γ for any spider γ. As a consequence, the height of a web $W(\alpha)$ is at most $|V(\alpha)|$.*

Each node γ of $W(\alpha)$ also has an associated value v_γ, which is defined by the following process:

- Initially, set $v_\alpha = 1$ and for all other γ, set $v_\gamma = 0$.
- Starting from the root and in topological order, each spider node γ adds $v_\gamma c_\beta$ to v_β for each child $\beta \in \mathcal{I}_\gamma$, where the c_β are the coefficients from Corollary 4.66.

Proposition 4.69. *If $\mathcal{M}$ satisfies the constraints "$\langle v, d_u\rangle^2 = 1$" and $x \perp \mathrm{Null}(\mathcal{M})$, then*

$$x^\intercal \left(M_\alpha - \sum_{\textit{leaves } \gamma \textit{ of } W(\alpha)} v_\gamma M_\gamma \right) x = 0.$$

Proof. Start with the equation $x^\intercal M_\alpha x = x^\intercal v_\alpha M_\alpha x$. In each step, we take the topologically first spider γ, which in this case means the spider closest to the root of $W(\alpha)$, that is present in the right hand side of our equation and using Corollary 4.66, we replace $v_\gamma M_\gamma$ by $\sum_{\beta \in \mathrm{children}(\gamma)} v_\gamma c_\beta M_\beta$. Precisely by the definition of the v_γ, this process ends with the equation

$$x^\intercal M_\alpha x = x^\intercal \left(\sum_{\text{leaves } \gamma \text{ of } W(\alpha)} v_\gamma M_\gamma \right) x$$

□

Proposition 4.70. *For any node β in $W(\alpha)$, $|\text{parents}(\beta)| \leq 4|V(\alpha)|^3 \cdot |E(\alpha)|^2$ where $parents(\beta)$ is the set of nodes γ in $W(\alpha)$ such that $\beta \in \mathcal{I}_\gamma$.*

Proof. The following process covers all parent left spiders γ which could possibly collapse their end vertices to form β. Starting from $\gamma = \beta$,

- Pick a circle vertex $\textcircled{u} \in V(\gamma)$ to be the neighbor of the end vertices.
- Pick a square vertex $\boxed{i} \in V(\gamma)$ to be the collapse of the first end vertex. "Uncollapse" it by adding a new square to U_γ with a single edge to $\textcircled{u}$ with label 1. Flip the value of $U_\gamma(\boxed{i})$. Modify the label of $\{\boxed{i}, \textcircled{u}\}$ to any number up to $|E(\alpha)|$.
- Pick a square vertex $\boxed{j} \in V(\gamma)$ to be the second end vertex. Optionally uncollapse it by adding a new square to γ in the same way as above.

The process can be carried out in at most $|V(\alpha)|^3|E(\alpha)|(|E(\alpha)| + 1) \leq 2|V(\alpha)|^3|E(\alpha)|^2$ ways. We multiply by 2 to accommodate right spiders. □

Let us label each parent-child edge (γ, β) as either a "type 1" edge if $\beta \in \mathcal{I}_\gamma^{(1)}$ or a "type 2" edge if $\beta \in \mathcal{I}_\gamma^{(2)}$.

Proposition 4.71. *Let p be a path in $W(\alpha)$ with $\#_1(p)$ type 1 edges and $\#_2(p)$ type 2 edges. Then $\#_1(p) \leq |E(\alpha)| + 2\#_2(p)$.*

Proof. For a shape γ, let S_γ be the set of square vertices in γ. Then, $S_\gamma \cap W_\gamma$ will be the set of middle vertices of γ which are squares. We claim that the quantity $\left|\mathcal{S}_\gamma \cap W_\gamma\right| + \left|U_\gamma \setminus (U_\gamma \cap V_\gamma)\right| + \left|V_\gamma \setminus (U_\gamma \cap V_\gamma)\right|$ decreases during a collapse.

Fix a pair of consecutive shapes (γ, β) which form a type 1 edge. Looking at the definition of $\mathcal{I}_\gamma^{(1)}$, each end vertex either collapses with (1) nothing, or (2) a vertex of W_γ, or (3) a vertex from $V_\gamma \setminus U_\gamma$ (if γ is a left spider; for a right spider, $U_\gamma \setminus V_\gamma$). Furthermore, case (2) or (3) must occur for at least one of the end vertices and also, they do not collapse together.

If case (2) occurs, then $\left|\mathcal{S}_\beta \cap W_\beta\right| < \left|\mathcal{S}_\gamma \cap W_\gamma\right|$ while $\left|U_\beta \setminus (U_\beta \cap V_\beta)\right| = \left|U_\gamma \setminus (U_\gamma \cap V_\gamma)\right|$ and $\left|V_\beta \setminus (U_\beta \cap V_\beta)\right| = \left|V_\gamma \setminus (U_\gamma \cap V_\gamma)\right|$. If case (3) occurs, then $W_\beta = W_\gamma$ while $\left|U_\beta \setminus (U_\beta \cap V_\beta)\right| < \left|U_\gamma \setminus (U_\gamma \cap V_\gamma)\right|$ and $\left|V_\beta \setminus (U_\beta \cap V_\beta)\right| < \left|V_\gamma \setminus (U_\gamma \cap V_\gamma)\right|$. In all cases, $\left|\mathcal{S}_\beta \cap W_\beta\right| + \left|U_\beta \setminus (U_\beta \cap V_\beta)\right| + \left|V_\beta \setminus (U_\beta \cap V_\beta)\right| < \left|\mathcal{S}_\gamma \cap W_\gamma\right| + \left|U_\gamma \setminus (U_\gamma \cap V_\gamma)\right| + \left|V_\gamma \setminus (U_\gamma \cap V_\gamma)\right|$ as desired.

Now we bound this expression for α. From the definition of $\mathcal{L}$, Definition 4.29, for spiders appearing in the pseudocalibration, the square vertices in W_α, $U_\alpha \setminus (U_\alpha \cap V_\alpha)$ and $V_\alpha \setminus (U_\alpha \cap V_\alpha)$ have degree at least 1 and can only be connected to circle vertices. Therefore their number is bounded by $|E(\alpha)|$. Hence, initially $|\mathcal{S}_\alpha \cap W_\alpha| + |U_\alpha \setminus (U_\alpha \cap V_\alpha)| + |V_\alpha \setminus (U_\alpha \cap V_\alpha)| \leq |E(\alpha)|$.

Finally, each type 2 edge in p can only increase $\left|\mathcal{S}_\gamma \cap W_\gamma\right| + \left|U_\gamma \setminus (U_\gamma \cap V_\gamma)\right| + \left|V_\gamma \setminus (U_\gamma \cap V_\gamma)\right|$ by at most 2. Therefore, we have the desired inequality $\#_1(p) \le |E(\alpha)| + 2\#_2(p)$. □

Corollary 4.72. $\#_2(p) \ge \frac{|p|}{3} - \frac{|E(\alpha)|}{3}$.

Proof. Plug in $|p| = \#_1(p) + \#_2(p)$ and rearrange. □

Finally, we can bound the accumulation on each non-spider by a term which only depends on the parameters of the spider α.

Lemma 4.73. *There are absolute constants C_1, C_2 so that for all leaves γ of $W(\alpha)$,*

$$\left|v_\gamma\right| \le (C_1 \cdot |V(\alpha)| \cdot |E(\alpha)|)^{C_2|E(\alpha)|}.$$

Proof. To bound $\left|v_\gamma\right|$ we will sum the contributions of all paths $p = (\beta_0 = \alpha, \dots, \beta_r = \gamma)$ in $W(\alpha)$ starting from α and ending at γ. This path contributes a product of coefficients c_β towards v_γ.

Remark 4.74. *Here it is important that type 2 edges have stronger bounds on their coefficients $\left|c_\beta\right| \le C \cdot (|V(\alpha)||E(\alpha)|)^{O(1)}/n \ll 1$.*

Before we proceed with the proof we establish some convenient notation and recall some facts. For consecutive shapes β_{i-1}, β_i (i.e. β_i is a child of β_{i-1}), we denote by c_{β_i} the coefficient from Corollary 4.66 applied on β_{i-1}. By Proposition 4.70, the in-degree of $W(\alpha)$ can be bounded as $B_1 \cdot (|V(\alpha)||E(\alpha)|)^{B_2}$ for some constants B_1, B_2. Thus, the number of paths of length r ending at γ is at most $(B_1|V(\alpha)||E(\alpha)|)^{B_2 r}$. Using Corollary 4.66, set B_1, B_2 large enough so that c_{β_i} is at most $B_1 \cdot (|V(\alpha)||E(\alpha)|)^{B_2}$ for a type 1 edge (resp. $B_1 \cdot (|V(\alpha)||E(\alpha)|)^{B_2}/n$ for a type 2 edge).

$$\begin{aligned}
|v_\gamma| &\le \sum_{r=0}^{\infty} \sum_{\substack{p=(\beta_0=\alpha,\dots,\beta_r=\gamma)\\ \text{path from } \alpha \text{ to } \gamma \text{ in } W(\alpha)}} \prod_{i=1}^{r} |c_{\beta_i}| \\
&\le \sum_{r=0}^{\infty} \sum_{\substack{p=(\beta_0=\alpha,\dots,\beta_r=\gamma)\\ \text{path from } \alpha \text{ to } \gamma \text{ in } W(\alpha)}} \left(B_1 \cdot (|V(\alpha)||E(\alpha)|)^{B_2}\right)^{\#_1(p)} \left(B_1 \cdot (|V(\alpha)||E(\alpha)|)^{B_2}/n\right)^{\#_2(p)} && \text{(Corollary 4.66)} \\
&\le \sum_{r=0}^{\infty} \sum_{\substack{p=(\beta_0=\alpha,\dots,\beta_r=\gamma)\\ \text{path from } \alpha \text{ to } \gamma \text{ in } W(\alpha)}} \left(B_1 \cdot (|V(\alpha)||E(\alpha)|)^{B_2}\right)^{|E(\alpha)|+2\#_2(p)} \left(B_1 \cdot (|V(\alpha)||E(\alpha)|)^{B_2}/n\right)^{\#_2(p)} && \text{(Proposition 4.71)} \\
&= \sum_{r=0}^{\infty} \sum_{\substack{p=(\beta_0=\alpha,\dots,\beta_r=\gamma)\\ \text{path from } \alpha \text{ to } \gamma \text{ in } W(\alpha)}} \left(B_1 \cdot (|V(\alpha)||E(\alpha)|)^{B_2}\right)^{|E(\alpha)|} \left(B_1' \cdot (|V(\alpha)||E(\alpha)|)^{B_2'}/n\right)^{\#_2(p)}
\end{aligned}$$

for some constants B_1', B_2'. We split the above sum into two sums, $r \le 3|E(\alpha)|$ and $r > 3|E(\alpha)|$. For $r \le 3|E(\alpha)|$, upper bounding the $\#_2(p)$ term by 1 and upper bounding the number of paths by $(B_1|V(\alpha)||E(\alpha)|)^{B_2 r}$ gives a bound of $(B_1''|V(\alpha)||E(\alpha)|)^{B_2''|E(\alpha)|}$ for some constants B_1'', B_2''. For larger r, we lower bound $\#_2(p) \geq r/9 = |E(\alpha)|/3$ using Corollary 4.72. Applying the same bound on the number of paths, the total contribution of the terms corresponding to larger r is bounded by 1 using the power of n in the denominator (assuming δ, τ are small enough). □

We define the result of all this spider killing to be a new matrix $\mathcal{M}^+$.

Definition 4.75. *Define the matrix $\mathcal{M}^+$ as the result of killing all the spiders,*

$$\mathcal{M}^+ := \mathcal{M} - \sum_{\textit{spiders } \alpha} \lambda_\alpha \left(M_\alpha - \sum_{\textit{leaves } \gamma \textit{ of } W(\alpha)} v_\gamma M_\gamma \right)$$

Let λ_α^+ be the coefficients in the graph matrix basis,

$$\mathcal{M}^+ = \sum_{\textit{shapes } \alpha} \lambda_\alpha^+ M_\alpha \,.$$

4.4.4 Finishing the proof

Proposition 4.76. *If β is a trivial shape, $\lambda_\beta^+ = \lambda_\beta$.*

Proof. A trivial shape cannot appear in $W(\alpha)$ for any α, since every collapse of a spider always keeps its circle vertices around. □

Lemma 4.77. *If β is a nontrivial non-spider and $\beta \in W(\alpha)$ for some spider $\alpha \in \mathcal{L}$, then*

$$|\lambda_\alpha| \left\| M_\beta \right\| \leq \frac{\eta^{|U_\beta| + |V_\beta|}}{n^{\Omega(\varepsilon |E(\alpha)|)}} \,.$$

Proof. This is the formal statement of Lemma 4.22. Using the formal norm bounds Lemma 4.46 and Lemma 4.47, the only unaccounted factor is log factor in the norm bound,

$$2 \cdot (|V(\beta)| \cdot (1 + |E(\beta)|) \cdot \log(n))^{C \cdot (|V_{rel}(\beta)| + |E(\beta)|)} \,.$$

The base is upper bounded by $n^{O(\tau)}$. In the exponent, we have $|V_{rel}(\beta)| \leq 2(|E(\alpha)| + |E(\beta)|)$ since all the degree 0 vertices in $V_{rel}(\beta)$ would have had vertices of $V_{rel}(\alpha)$ collapse into it in the chain of collapses and there are no degree 0 vertices in $V_{rel}(\alpha)$. Finally, since $|E(\beta)| \leq |E(\alpha)|$, the exponent is at most $O(|E(\alpha)|)$. Therefore the log factor can be absorbed into $\frac{1}{n^{\Omega(\varepsilon |E(\alpha)|)}}$. □

Lemma 4.78. *For $k, l \in \{0, 1, \ldots, D/2\}$, let $\mathcal{B}_{k,l}$ denote the set of nontrivial non-spiders on block (k, l). Then*

$$\sum_{\beta \in \mathcal{B}_{k,l}} \left| \lambda_\beta^+ \right| \left\| M_\beta \right\| \leq \eta^{k+l} \cdot \frac{1}{n^{\Omega(\varepsilon)}}$$

Proof.

$$\sum_{\beta \in \mathcal{B}_{k,l}} \left\| \lambda_\beta^+ M_\beta \right\| \leq \sum_{\beta \in \mathcal{B}_{k,l}} \left| \lambda_\beta \right| \left\| M_\beta \right\| + \sum_{\beta \in \mathcal{B}_{k,l}} \sum_{\substack{\text{spiders } \alpha: \\ \beta \in W(\alpha)}} \left| v_\beta \right| \left| \lambda_\alpha \right| \left\| M_\beta \right\|$$

To bound the first term, we checked previously in Corollary 4.50 that the total norm of nontrivial non-spiders appearing in the pseudocalibration (i.e. this term) is $\eta^{k+l} o_n(1)$. For the second term, via Lemma 4.73 we have a bound on the accumulations v_γ of one spider on one non-spider, so it is at most

$$\leq \sum_{\beta \in \mathcal{B}_{k,l}} \sum_{\substack{\text{spiders } \alpha: \\ \beta \in W(\alpha)}} (C_1 |V(\alpha)| \cdot |E(\alpha)|)^{C_2 |E(\alpha)|} \cdot |\lambda_\alpha| \left\| M_\beta \right\|.$$

Invoking the charging argument for non-spiders which are collapses, Lemma 4.77,

$$\begin{aligned} &\leq \eta^{k+l} \cdot \sum_{\beta \in \mathcal{B}_{k,l}} \sum_{\substack{\text{spiders } \alpha: \\ \beta \in W(\alpha)}} \left(\frac{C_1 |V(\alpha)| \cdot |E(\alpha)|}{n^{\Omega(\varepsilon)}} \right)^{C_2' |E(\alpha)|} \\ &\leq \eta^{k+l} \cdot \sum_{\beta \in \mathcal{B}_{k,l}} \sum_{\substack{\text{spiders } \alpha: \\ \beta \in W(\alpha)}} \left(\frac{C_1 n^\tau \cdot n^\tau}{n^{\Omega(\varepsilon)}} \right)^{C_2' |E(\alpha)|}. \end{aligned}$$

Bound the sum over all spiders by the sum over all shapes. By Proposition 4.49, the number of shapes with i edges is $n^{O(\tau(i+1))}$. Summing by the number of edges, observe that $|E(\alpha)| \geq \max(|E(\beta)|, 2)$ since spiders always have at least 2 edges.

$$\begin{aligned} &\leq \eta^{k+l} \sum_{\beta \in \mathcal{B}_{k,l}} \sum_{i = \max(|E(\beta)|, 2)}^{\infty} n^{O(\tau(i+1))} \cdot \left(\frac{C_1 n^\tau \cdot n^\tau}{n^{\Omega(\varepsilon)}} \right)^{C_2' i} \\ &\leq \eta^{k+l} \sum_{\beta \in \mathcal{B}_{k,l}} \frac{1}{n^{\Omega(\varepsilon \max(|E(\beta)|, 2))}} \\ &\leq \eta^{k+l} \sum_{i=0}^{\infty} \frac{n^{O(\tau(i+1))}}{n^{\Omega(\varepsilon \max(i, 2))}} \\ &= \eta^{k+l} \cdot \frac{1}{n^{\Omega(\varepsilon)}} \end{aligned}$$

□

Corollary 4.79. *For $k \in \{0, \ldots, D/2\}$, the (k,k) block of $\mathcal{M}^+$ has minimum singular value at least $\eta^{2k}(1 - \frac{1}{n^{\Omega(\varepsilon)}})$, and for $k, l \in \{0, \ldots, D/2\}, l \neq k$, the (k,l) off-diagonal block has norm at most $\eta^{k+l} \cdot \frac{1}{n^{\Omega(\varepsilon)}}$.*

Proof. By Proposition 4.76 the identity matrix appears on the (k,k) blocks with coefficient η^{2k}. By construction, $\mathcal{M}^+$ has no spider shapes. By Lemma 4.78, the total norm of the non-spider shapes on the (k,l) block is at most $\eta^{k+l} \cdot \frac{1}{n^{\Omega(\varepsilon)}}$. □

This implies that $\mathcal{M}^+$ is PSD (in fact, positive definite). To conclude that $\mathcal{M}$ is PSD, we need to fix $\mathcal{M}$ so that it exactly satisfies the constraints "$\langle v, d_u \rangle^2 = 1$", rather than approximately.

Lemma 4.80. *There is an* $\binom{[n]}{\leq D/2} \times \binom{[n]}{\leq D/2}$ *matrix* $\mathcal{E}$ *with* $\|\mathcal{E}\| \leq \frac{1}{n^{\Omega(n^\tau)}}$ *such that the matrix* $\mathcal{M}_{fix} := \mathcal{M} + \mathcal{E}$ *is SoS-symmetric and exactly satisfies the constraints "*$\langle v, d_u \rangle^2 = 1$*".*

Proof. This is the statement of [GJJ$^+$20, Corollary 7.3]. The proof is omitted. □

Theorem 4.81. *W.h.p.* $\mathcal{M}_{fix} \succeq 0$.

Proof. For any $x \in \mathrm{Null}(\mathcal{M}_{fix})$, we of course have $x^\intercal \mathcal{M}_{fix} x = 0$. For any $x \perp \mathrm{Null}(\mathcal{M}_{fix})$ with $\|x\|_2 = 1$,

$$\begin{aligned}
x^\intercal \mathcal{M}_{fix} x &= x^\intercal (\mathcal{M} + \mathcal{E}) x \\
&= x^\intercal \mathcal{M}^+ x + x^\intercal \left(\sum_{\text{spiders } \alpha} \lambda_\alpha \left(\mathcal{M}_\alpha - \sum_{\text{leaves } \gamma \text{ of } W(\alpha)} v_\gamma M_\gamma \right) \right) x \\
&\quad + x^\intercal \mathcal{E} x \\
&= x^\intercal (\mathcal{M}^+ + \mathcal{E}) x && \text{(Proposition 4.69)}
\end{aligned}$$

Because the norm bound on $\mathcal{E}$ in Lemma 4.80 is significantly less than $\eta^D = n^{-n^\delta}$, the bound on the norm of each block of $\mathcal{M}^+$ in Corollary 4.79 also applies to the blocks of $\mathcal{M}^+ + \mathcal{E}$. Therefore, we use Proposition 4.44 to conclude $\mathcal{M}^+ + \mathcal{E} \succeq 0$ and the above expression is nonnegative. □

4.5 Reduction from SK model to PAP

Here, we prove Theorem 4.5 and Theorem 4.2.

Recall that in the Planted Boolean Vector problem, we wish to optimize

$$\mathsf{OPT}(V) := \frac{1}{n} \max_{b \in \{\pm 1\}^n} b^\intercal \Pi_V b,$$

where V is a uniformly random p-dimensional subspace of $\mathbb{R}^n$.

Theorem 4.5. *There exists a constant* $c > 0$ *such that, for all* $\varepsilon > 0$ *and* $\delta \leq c\varepsilon$*, for* $p \geq n^{2/3+\varepsilon}$*, w.h.p. over* V *there is a degree-*n^δ *SoS solution for Planted Boolean Vector of value* 1.

Proof. We wish to produce an SoS solution $\widetilde{\mathbb{E}}$ on Boolean variables $b_1, \dots, b_n$ such that $\widetilde{\mathbb{E}}[b^\intercal \Pi_V b] = n$. Instead of sampling a uniformly random p-dimensional subspace V of $\mathbb{R}^n$, we first sample $d_1, \dots, d_n$ i.i.d. p-dimensional Gaussian vectors from $\mathcal{N}(0, I)$, then form an n-by-p matrix A with rows $d_1, \dots, d_n$, and finally take V to be the span of the columns of A. Since the columns of A are isotropic i.i.d. random Gaussian vectors, we have that V is a uniform p-dimension subspace[5] of $\mathbb{R}^n$.

We will consider V as the input for the Planted Boolean Vector problem while the vectors $d_1, \dots, d_n$ will be used to construct a pseudoexpectation operator for the Planted Affine Planes problem[6]. Since $n \le p^{3/2-\Omega(\epsilon)}$, by Theorem 4.4, for all $\delta \le c\epsilon$ for a constant $c > 0$, w.h.p., there exists a degree-n^δ pseudoexpectation operator $\widetilde{\mathbb{E}}'$ on formal variables $v = (v_1, \dots, v_p)$ such that $\widetilde{\mathbb{E}}'[\langle v, d_u\rangle^2] = 1$ for every $u \in [n]$.

Define $\widetilde{\mathbb{E}}$ by $\widetilde{\mathbb{E}}[b_u] := \widetilde{\mathbb{E}}'[\langle v, d_u\rangle]$ for all $u \in [n]$ and extending it to all polynomials on $\{b_u\}$ by multilinearity. This is well defined because $\widetilde{\mathbb{E}}'[\langle v, d_u\rangle^2] = 1$. Note that $\widetilde{\mathbb{E}}$ is a valid pseudoexpectation operator of the same degree as $\widetilde{\mathbb{E}}'$. Finally, observe that

$$\frac{1}{n}\widetilde{\mathbb{E}}[b^\intercal \Pi_V b] = \frac{1}{n}\widetilde{\mathbb{E}}'[v^\intercal A^\intercal \Pi_V A v] = \frac{1}{n}\widetilde{\mathbb{E}}'[v^\intercal A^\intercal A v] = 1.$$

□

Now we prove lower bounds for the Sherrington-Kirkpatrick problem, using a reduction and proof due to [MRX20]. We include it here for completeness. Recall that the SK problem is to compute

$$\mathsf{OPT}(W) := \max_{x \in \{\pm 1\}^n} x^\intercal W x,$$

where W is sampled from $\mathrm{GOE}(n)$.

Theorem 4.2. *There exists a constant $\delta > 0$ such that, w.h.p. for $W \sim \mathrm{GOE}(n)$, there is a degree-n^δ SoS solution for the Sherrington–Kirkpatrick problem with value at least $(2 - o(1)) \cdot n^{3/2}$.*

We will use the following standard results from random matrix theory of $\mathrm{GOE}(n)$.

Fact 4.82. *Let $\lambda_1 \ge \ldots \ge \lambda_n$ be the eigenvalues of $W \sim \mathrm{GOE}(n)$ with corresponding normalized eigenvectors $w_1, \dots, w_n$. Then,*

1. *For every $p \in [n]$, the span of $w_1, \dots, w_p$ is a uniformly random p-dimensional subspace of $\mathbb{R}^n$ (see e.g. [OVW16, Section 2]).*
2. *W.h.p., $\lambda_{n^{0.67}} \ge (2 - o(1))\sqrt{n}$ (Corollary of Wigner's semicircle law [Wig93])*

Proof of Theorem 4.2. Let $p = n^{0.67}$ and $W \sim \mathrm{GOE}(n)$. Let $\lambda_1 \ge \ldots \ge \lambda_n$ be the eigenvalues of W with corresponding orthonormal set of eigenvectors $w_1, \dots, w_n$. By Fact 4.82, we

[5] Except for a zero measure event.

[6] Note that the vectors d_u are not "given" in the Planted Boolean Vector problem, though the construction of $\widetilde{\mathbb{E}}$ is not required to be algorithmic in any sense anyway.

have that $\lambda_p \geq (2 - \mathrm{o}(1))\sqrt{n}$ and that $w_1, \ldots, w_p$ span a uniformly random p-dimensional subspace V of $\mathbb{R}^n$.

We consider V as the input of the Boolean Planted Vector problem and by Theorem 4.5, for some constant $\delta > 0$, w.h.p. there exists a degree-n^δ pseudoexpectation operator $\widetilde{\mathbb{E}}$ such that $\widetilde{\mathbb{E}}[x_i^2] = 1$ and $\widetilde{\mathbb{E}}[\sum_{i=1}^p \langle x, w_i \rangle^2] = \widetilde{\mathbb{E}}[x^\intercal \Pi_V x] = n$. Now,

$$\begin{aligned}\widetilde{\mathbb{E}}[x^\intercal W x] = \widetilde{\mathbb{E}}[\sum_{i=1}^n \lambda_i \langle x, w_i \rangle^2] &\geq \lambda_p \widetilde{\mathbb{E}}[x^\intercal \Pi_V x] - |\lambda_n| \widetilde{\mathbb{E}}[\sum_{i=p+1}^n \langle x, w_i \rangle^2] \\ &\geq (2 - \mathrm{o}(1)) n^{3/2} - |\lambda_n| \widetilde{\mathbb{E}}[\langle x, x \rangle - \sum_{i=1}^p \langle x, w_i \rangle^2] \\ &= (2 - \mathrm{o}(1)) n^{3/2}.\end{aligned}$$

□

Remark 4.83. *Using the same proof as above, we can obtain Theorem 4.2 even if we were only able to prove SoS lower bounds for Planted Affine Planes for some $m = \omega(n)$. So, pushing the value of m up to $n^{3/2-\epsilon}$, which is Theorem 4.4, offers only a modest improvement.*

4.6 Open Problems

In light of the result of Montanari [Mon21], the situation is intriguing. Montanari showed that for all $\varepsilon > 0$, there is a $\mathrm{O}_\varepsilon(n^2)$ time randomized algorithm that given a random W drawn from the Gaussian Orthogonal Ensemble, outputs an x such that $x^T W x \geq (1 - \varepsilon)\mathsf{OPT}(W)$. The correctness of the algorithm assumes a widely-believed conjecture from statistical physics known as the full replica symmetry breaking assumption. However, we show an integrality gap for SoS.

Based on this, it is an interesting question whether SoS, together with an appropriate rounding scheme, is optimal for the Sherrington-Kirkpatrick problem. On the one hand, the situation could be similar to the Feige-Schechtman integrality gap instance for MaxCut [FS02]. For the Feige-Schechtman integrality gap instance, SoS fails to certify the value of the optimal solution. However, applying hyperplane rounding to the SoS solution gives an almost-optimal solution for these instances. It could be the case that there is a rounding scheme which takes an SoS solution for the Sherrington-Kirkpatrick problem on a random W and returns an almost optimal solution x. On the other hand, we currently don't know what this rounding scheme would be.

We conjecture that the Planted Affine Planes problem remains difficult even with the number of vectors increased to $m = n^{2-\epsilon}$.

Conjecture 4.84. *Theorem 4.4 holds with the bound on the number of sampled vectors m loosened to $m \leq n^{2-\varepsilon}$.*

The reason for the upper bound comes from Remark 4.20. Analyzing $\widetilde{\mathbb{E}}[1]$ is an established way to hypothesize about the power of SoS in hypothesis testing problems (see

[Hop18, HKP+17]).

Dual to the Planted Affine Planes problem, we conjecture a similar bound for Planted Boolean Vector problem whenever $d \geq n^{1/2+\varepsilon}$.

Conjecture 4.85. *Theorem 4.5 holds with the bound on the dimension p of a random subspace loosened to $p \geq n^{1/2+\varepsilon}$.*

The generic quadratic optimization problem $\max_{x\in\{-1,+1\}^n} x^\intercal W x$ does not have a $\log^\gamma n$-approximation algorithm, assuming $\mathsf{NP} \not\subseteq \mathsf{quasi\text{-}P}$ [ABE+05]. However, in the average-case setting, there is only a constant-factor integrality gap for the spectral algorithm and also for sum-of-squares. Can we prove $\log^{\Omega(1)} n$ integrality gap for sum-of-squares running on a different instance?

Certifying bounds on $x^\intercal W x$ is also of interest when W is a random sparse matrix. When W is the adjacency matrix of a random d-regular graph, this is the MaxCut problem. An SoS lower bound is conjectured for this problem:

Conjecture 4.86. *Let $d \geq 3$, and let G be a random d-regular graph on n vertices. For some $\delta > 0$, w.h.p. there is a degree-n^δ pseudoexpectation operator $\widetilde{\mathbb{E}}$ on boolean variables x_i with MaxCut value at least*

$$\frac{1}{2} + \frac{\sqrt{d-1}}{d}(1 - o_{d,n}(1))$$

The above expression is w.h.p. the value of the spectral relaxation for MaxCut, therefore qualitatively this conjecture expresses that degree n^δ SoS cannot significantly tighten the basic spectral relaxation (conversely, the true MaxCut value is near $\frac{1}{2} + \frac{P^*}{\sqrt{d}}$ where P^* is the Parisi constant [DMS17]). Proving this conjecture is likely to require techniques for the sparse regime developed in Chapter 6.

CHAPTER 5
INNER PRODUCT POLYNOMIALS

In this chapter, we investigate a Fourier-like basis for a particular function space, the space of *orthogonally-invariant* functions of a collection of random vectors $\{d_u\}$. These functions only depend on the "angular configuration" of the d_u, i.e. how the vectors are arranged in space, and not on the entries of the vectors themselves.

We give an "almost-orthogonal" polynomial basis for this vector space when the random vectors are Gaussian, spherical, or Boolean. Specifically, polynomials of different degree will be orthogonal, and polynomials with the same degree will have inner product that is small relative to n, the dimension of the random vectors. In all three cases, our basis admits an interesting combinatorial description based on the topology of an underlying graph.

We encountered the p_G basis in the course of the work [GJJ$^+$20] described in Chapter 4. There, we construct a matrix $\mathcal{M}$ which is essentially an orthogonally-invariant function of a collection of random Gaussian vectors $\{d_u\}$; the entries of $\mathcal{M}$ are naturally expressed (via "pseudocalibration") in terms of an orthogonal polynomial basis evaluated on the d_u. Ultimately, we ended up using the standard Hermite basis as this was sufficient for our purposes, though we also considered using the p_G basis.

We expect that the p_G will be most useful for applications where we work with large n and relatively low-degree moments of the random vectors, such as analyzing the sum-of-squares hierarchy or rounding algorithms for semidefinite programs. While other bases may be simpler, the p_G basis is specialized to orthogonally invariant functions and it exhibits nontrivial combinatorial cancellations which would be hard to spot and explain in other bases, and which might be intrinsic to some problems.

In Section 5.1 we give a general overview of the polynomials, related work, and some general properties. The three cases of Gaussian, spherical, and Boolean random vectors are treated separately in Section 5.2, Section 5.3, Section 5.4. The spherical case exhibits some intricacies. We show that the inner product between two polynomials is always nonnegative in the Gaussian and Boolean cases. In the spherical case, we show that it can be negative, but we conjecture that if the underlying graph of inner products is planar then the inner product will always be non-negative. We prove an approximate Fourier inversion formula using the combinatorial properties of the basis in Section 5.5.

Bibliography. This chapter is the content of [JP22]. The relationship to the Wick product in Section 5.1.1 and Section 5.2 is new. The reader may benefit from the omitted table of example polynomials in [JP22, Appendix A].

5.1 Overview

When we have a collection of random variables, it is often extremely useful to find a basis of polynomials in the random variables which is orthonormal under the natural inner product $\langle f, g \rangle := \mathbb{E}[f \cdot g]$, as demonstrated throughout the other chapters of this thesis. Some important examples are as follows:

1. If x is a random point of the Boolean hypercube $\{-1, 1\}^n$ then the multilinear monomials $\{\prod_{i \in S} x_i : S \subseteq [n]\}$ are an orthonormal basis.

2. When we have a single Gaussian variable $x \sim \mathcal{N}(0, 1)$, the Hermite polynomials (with the correct normalization) are an orthonormal basis. When x is an n-dimensional vector with Gaussian coordinates (i.e. $x \sim \mathcal{N}(0, \mathrm{Id}_n)$), the multivariate Hermite polynomials form an orthonormal basis.

3. When $x \in \mathbb{R}^n$ is a random unit vector (i.e. $x \in_{\mathrm{R}} S^{n-1}$), spherical harmonics give an orthonormal basis.

In this chapter, we consider polynomials of inner products between a collection of random vectors. More precisely, fix a finite set of vertices V and $n \in \mathbb{N}$ and consider drawing i.i.d. random n-dimensional vectors d_u for each $u \in V$. We will work in three settings: when the n-dimensional vector d_u is a standard Gaussian, a uniform unit vector, and a uniform Boolean vector. We consider polynomials in the variables $d_{u,i}$ with real coefficients which have degree less than n and are orthogonally invariant i.e. unchanged if the $\{d_u\}$ are simultaneously replaced by $\{Td_u\}$ for any orthogonal matrix T. Any such orthogonally invariant polynomial will also be expressible[1] in terms of the inner product variables $x_{uv} := \langle d_u, d_v \rangle$.

A natural spanning set for the space of orthogonally invariant polynomials is the set of monomials $\prod_{u,v \in V} x_{uv}^{k_{uv}}$ where each $k_{uv} \in \mathbb{N}$. Equivalently, there is one monomial for each undirected multigraph on V (with self-loops allowed in the Gaussian case): for the monomial $\prod_{u,v \in V} x_{uv}^{k_{uv}}$ we take the graph where there are k_{uv} multi-edges from u to v. We denote this monomial by m_G where G is the underlying graph.

However, the monomials m_G are not orthogonal. For example, one can check that in the Gaussian case, the graph shown in Fig. 5.1 has

$$\mathbb{E}[x_{12}x_{23}x_{34}x_{14}] = \mathop{\mathbb{E}}_{d_1, d_2, d_3, d_4 \sim \mathcal{N}(0, \mathrm{Id}_n)} [\langle d_1, d_2 \rangle \langle d_2, d_3 \rangle \langle d_3, d_4 \rangle \langle d_1, d_4 \rangle] = n.$$

Our goal in this work is to orthogonalize the m_G into a basis of polynomials p_G. As it turns out, the basis p_G which we will obtain is not quite orthogonal, but it is very close. In particular, we will have that $\langle p_G, p_H \rangle = 0$ unless $V(G) = V(H)$ and G and H have the same degree at every vertex. In addition, even when $G \neq H$ and $\langle p_G, p_H \rangle \neq 0$, $\langle p_G, p_H \rangle$ will be small (see Lemma 5.89).

While the p_G basis is not quite orthogonal, it exhibits some surprisingly beautiful combinatorics based on the underlying graph G. Even computing $\mathbb{E}[m_G]$, one can already see

[1] When d_u is a Boolean vector, we instead require that the polynomials are invariant under permutations of $[n]$ and changing the signs of coordinates (i.e. automorphisms of the Boolean hypercube). In this setting, in addition to inner products, we also have k-wise inner products for all even $k > 2$. For more details, see Section 5.4.

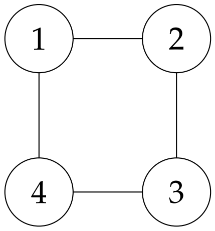


Figure 5.1

a connection to the topology of the graph G. In the Gaussian case, the magnitude of $\mathbb{E}[m_G]$ is n^k where k is the maximum number of cycles that $E(G)$ can be partitioned into (and is 0 if G has a vertex with odd degree) and analogous results hold for the spherical and Boolean cases (see Lemma 5.19, Lemma 5.41, and Lemma 5.74). A theme of this chapter is that quantities involving the m_G and p_G may not have clean exact formulas, but their magnitudes in n are determined by combinatorial and topological properties of G (these combinatorial properties are usually NP-hard to compute from G).

5.1.1 Constructing the polynomials

Given any inner product on polynomials, we can automatically construct an orthonormal basis of polynomials by using the Gram-Schmidt process. However, to run Gram-Schmidt, it is necessary to choose an order. A natural order for polynomials is by degree, though within each degree it is not clear how the polynomials should be ordered. We skirt this issue by only orthogonalizing a monomial against polynomials with lower degree[2]. The resulting polynomials we produce are "mostly orthogonal", with $\mathbb{E}[p_G \cdot p_H]$ possibly nonzero for polynomials of the same degree (in fact, they will be orthogonal unless G and H have the same degree on every vertex). We call this the *degree-orthogonal Gram-Schmidt process*.

Definition 5.1. *A polynomial family $\{p_I\}_{I \in \mathcal{I}}$ is degree-orthogonal (with respect to $\mathcal{D}$) if $\mathbb{E}_{d \sim \mathcal{D}}[p_I(d) p_J(d)] = 0$ whenever $\deg(p_I) \neq \deg(p_J)$.*

The degree-orthogonal Gram-Schmidt process outputs the unique monic degree-orthogonal basis.

Fact 5.2 (Uniqueness of degree-orthogonal basis). *Let $\{m_I\}_{I \in \mathcal{I}}$ be the set of monomials of degree at most τ in a set of variables v and let $\mathcal{D}$ be a distribution on $\mathbb{R}^v$ such that $\{m_I\}_{I \in \mathcal{I}}$ are linearly independent as functions on the support of $\mathcal{D}$. There is a unique set of monic polynomials $\{p_I\}_{I \in \mathcal{I}}$ such that*

(i) The unique monomial of maximum degree in p_I is m_I,

(ii) The family p_I is degree-orthogonal with respect to $\mathcal{D}$.

Furthermore, the p_I are linearly independent and span the same space as the m_I.

[2] Degree of a polynomial in this chapter always refers to total degree.

Proof. Condition (i) says that p_I lies in the space $\operatorname{span}(\{m_I\} \cup \{m_J : \deg(m_J) < \deg(m_I)\})$. Condition (ii) says that p_I is orthogonal to the latter subspace of codimension 1, and therefore p_I is determined since it's monic. □

Remark 5.3. *Our monomials $\{m_G\}$ are not linearly independent when the degree is too high. In this case, $\{p_G\}$ will be a spanning set rather than a basis.*

However, Gram-Schmidt certainly does not guarantee any nice description of the resulting polynomials. It turns out that the p_G also have closed-form combinatorial descriptions and we now give one such description. However, calculations are still a pain using this description. In the next sections we will give alternate combinatorial formulas for the p_G based on collections of matchings that allow for calculations, and also highlight the connection between the p_G and the topology of the graph G.

An equivalent view of the degree-orthogonal Gram-Schmidt process is as follows. We start with the vector space of polynomials $P_D = \mathbb{R}^{\leq D}[d_{1,1}, d_{1,2}, \dots, d_{1,n}, d_{2,1}, \dots,]$ of degree at most D, endowed with the expectation inner product. Gram-Schmidt decomposes P_D as:

$$P_D = \bigoplus_{i=0}^{D} H_i$$

where each H_{i+1} is the orthogonal complement in P_{i+1} of the smaller-dimensional spaces,

$$H_{i+1} = P_{i+1} \ominus \left(\bigoplus_{j=1}^{i} H_j \right) .$$

The spaces H_i are definitionally degree-orthogonal. Therefore, a tautological way to produce the degree-orthogonal basis is to project a basis m_G to $\pi_{\deg(m_G)}(m_G)$, where π_i is the projection operator to H_i.[3]

The point is that the projection operators are reasonably explicit. Whenever the underlying random variables are jointly Gaussian, the projection π_i is the *Wick product* and the decomposition into $\oplus_i H_i$ is known as the *Wiener chaos decomposition* [Jan97]. We will describe the Wick product explicitly in Section 5.2. If the underlying random variables are independent Booleans, recall that a Boolean Fourier character (i.e., a multilinear monomial) is already orthogonal to polynomials of lower degree. Therefore the projection in the Boolean case is the identity (and also zeroing out non-multilinear terms).

For general vectors d_u drawn i.i.d from a distribution $\mathcal{D}$ on $\mathbb{R}^n$, the projection operator is as follows. Let $\{\chi_\alpha : \alpha \in \mathbb{N}^n\}$ be the monic polynomial family on $\mathbb{R}^n$ which is degree-orthogonal under $\mathcal{D}$ (this is the set of degree-orthogonal polynomials for a single vector, e.g. the Hermite polynomials in the Gaussian case. This family is unique by Fact 5.2). Then:

[3] It's not clear that projection preserves orthogonal invariance of m_G, but we will check that this is the case in our settings.

Proposition 5.4. *Let $\prod_{u\in V} d_u^{\alpha_u}$ be a monomial, $\alpha_u \in \mathbb{N}^n$. Let $D = \sum_{u\in V}|\alpha_u|$ be the degree of the monomial. The projection π_k is:*

$$\pi_k\left(\prod_{u\in V} d_u^{\alpha_u}\right) = \begin{cases} \prod\limits_{u\in V} \chi_{\alpha_u}(d_u) & k = D \\ 0 & k \neq D\,. \end{cases}$$

In words, the projection replaces each monomial by the $\mathcal{D}$-orthogonal polynomial with that leading monomial.

Proof. First, we show that the result is in H_D i.e. it is orthogonal to all polynomials of lower degree. Let $m = \prod_{u\in V} d_u^{\beta_u}$ be a monomial with $\deg(m) < D$. Using independence of the d_u,

$$\mathbb{E}\left[m \cdot \prod_{u\in V} \chi_{\alpha_u}(d_u)\right] = \prod_{u\in V} \mathbb{E}\left[d^{\beta_u} \cdot \chi_{\alpha_u}(d_u)\right]\,.$$

Since $\deg(m) < D$, one of the degrees $|\beta_u|$ must be smaller than the corresponding degree $|\alpha_u|$. When the expectation is taken over just this vector d_u, by orthogonality of the χ_α, the expectation is zero.

Second, since the degree-D part is not changed by this map (on the right-hand side it remains the unique monomial $\prod_{u\in V} d_u^{\alpha_u}$, because χ_{α_u} is monic), the projection is the identity on degree-D polynomials modulo polynomials of degree $<D$. □

5.1.2 Related work

Although Gram-Schmidt works well for univariate polynomials, in general finding an *explicit* orthogonal basis of polynomials for a given space is a difficult task. Examples include polynomials on the unit ball and simplex [DX13] or a slice of the hypercube [Fil16]. Occasionally it is simpler to find a degree-orthogonal family, as we do here. For example, "the" spherical harmonics (as originally given by Laplace in $n = 3$ dimensions, see Chapter 4 of [DX14] for general n) are an orthogonal basis for functions on the sphere. However, it is easier to use the "Maxwell representation", which is only degree-orthogonal, as we do in Section 5.3.

In the Gaussian case, the basis p_G can be computed using the *Wick product* [Jan97]. This connection is explained further in Section 5.2.

To the best of our knowledge, the p_G have not been explicitly explored before. We now compare the p_G with several similar families of polynomials.

Some of the combinatorics of the monomials m_G is captured by the *circuit partition polynomial* [Bol02] (see also the *Martin polynomial* [Mar77, EM98]) which is the univariate generating function for circuit partitions of G:

$$r_G(x) = \sum_{k\geq 0} r_k(G)x^k$$

where $r_k(G)$ is the number of ways to split the edges of G into exactly k circuits. $r_G(n) = \mathbb{E}[m_G]$ for the Gaussian distribution, as we show in Lemma 5.19. This formula was also computed by Moore and Russell [MR10], who also prove the spherical case, Lemma 5.41.

When G equals k multiedges between two vertices 1 and 2, p_G generalizes a univariate orthogonal polynomial family evaluated on $\langle d_1, d_2 \rangle$. For the spherical case this is the Gegenbauer polynomials. For the Boolean case, this is the Kravchuk polynomials (after an affine shift). For the Gaussian case, p_G also depends on $\|d_1\|$ and $\|d_2\|$, but evaluated on $\langle d_1, d_1 \rangle = \langle d_2, d_2 \rangle = n$ this is the (probabilist's) Hermite polynomials.

The *matching polynomial* of a graph G is the univariate generating function for the number of matchings in G. Despite both families generalizing e.g. the Hermite polynomials, the matching polynomials and p_G seem incomparable.

For a permutation group $G \le S_k$, one defines the *cycle polynomial* [CS18]

$$\sum_{g \in G} x^{\text{number of cycles in } g}.$$

Though this is similar in appearance to some calculations in this paper, there is not a clear group G associated with the matching structures that we consider.

A possible approach to constructing an orthogonal basis for symmetric functions $f : (\mathbb{R}^n)^V \to \mathbb{R}$ is to take the "usual" orthogonal basis and symmetrize it. For example, the Hermite polynomial $h_\alpha(d_1, \dots, d_V)$, $\alpha \in \mathbb{N}^{n \times V}$ can be symmetrized into an orthogonally invariant function by computing:

$$\mathop{\mathbb{E}}_{T \in_{\mathrm{R}} O(n)} [h_\alpha(Td_1, \dots, Td_V)] .$$

However, this does *not* produce the same polynomials p_G that we construct in Section 5.2. Our polynomials appear simpler.

5.2 Polynomial Basis for the Gaussian Setting

In this section we investigate the family $\{p_G\}$ when $d_u \sim \mathcal{N}(0, \mathrm{Id}_n)$ i.i.d. The graph G is a multigraph on V, possibly with self-loops. We will develop a combinatorial understanding of the polynomials through "routings" (Lemma 5.14) and use it to give formulas for the inner product (Lemma 5.26) and variance (Corollary 5.28). We start by explaining the Wick product, which generally degree-orthogonalizes any function of Gaussian random variables, and then we specialize to our setting.

5.2.1 Wick Calculus

For functions on Gaussian space, nice combinatorial formulas for orthogonal polynomials arise via the *Wick product*.

Definition 5.5 (Wick product)**.** *Given jointly Gaussian random variables $X_1, \dots, X_N$, the Wick product is*

$$:X_1 \cdots X_N: = \sum_{\substack{\text{matchings } M \\ \text{on } [N]}} (-1)^{|M|} \prod_{(u,v) \in M} \mathbb{E}[X_u X_v] \prod_{\text{unmatched } u \in [N]} X_u \,.$$

Importantly, the X_i do not need to be independent or standard Gaussians. Some of the X_i may be linear combinations of or the same as other X_i. For example:

Proposition 5.6. *If $X = X_1 = \cdots = X_N$ are copies of a standard Gaussian random variable, then*

$$:X^n: = h_n(X)$$

where h_n is the n-th Hermite polynomial.

The Wick product orthogonalizes a monomial against lower degree. Extending it by linearity, it can be used to orthogonalize a polynomial against lower degree.

Proposition 5.7. *For jointly Gaussian random variables $X_1, \dots, X_N$ and a homogeneous polynomial $p(X_1, \dots, X_N)$ of degree D, the Wick product $:p:$ is orthogonal to any polynomial in $X_1, \dots, X_N$ with degree less than D.*

Lemma 5.8. *If jointly Gaussian random variables $X_1, \dots, X_M$ are independent (uncorrelated) from $Y_1, \dots, Y_N$, then*

$$:X_1 \cdots X_M Y_1 \cdots Y_N: = :X_1 \cdots X_M::Y_1 \cdots Y_N:$$

Proof. In Definition 5.5, any matching that goes between the X and Y variables is multiplied by $\mathbb{E}[X_i Y_j] = 0$. Therefore, the sum factors into matchings on $X_1, \dots, X_M$ and matchings on $Y_1, \dots, Y_N$. □

5.2.2 Definitions of inner product polynomials

We will give several equivalent formulas for the orthogonalized polynomials p_G. First, use the Wick product to orthogonalize the m_G.

Definition 5.9 (Wick product definition of p_G)**.** $p_G = :m_G:$

Lemma 5.10 (Hermite sum definition of p_G)**.**

$$p_G = \sum_{\sigma : E(G) \to [n]} \prod_{u \in V, i \in [n]} h_{|\{e \ni u \,:\, \sigma(e) = i\}|}(d_{u,i}).$$

Note that in this definition we consider a self-loop at u labeled i to contribute 2 to $|\{e \ni u \,:\, \sigma(e) = i\}|$.

Proof of Lemma 5.10.

$$m_G = \sum_{\sigma:E(G)\to[n]} \left(\prod_{\{u,v\}\in E(G)} d_{u,i}d_{v,i} \right)$$

$$:m_G: = \sum_{\sigma:E(G)\to[n]} : \left(\prod_{\{u,v\}\in E(G)} d_{u,i}d_{v,i} \right) :$$

Since the $d_{u,i}$ are independent, we may use Lemma 5.8 to move the Wick product onto each $d_{u,i}$ separately. Then, by Proposition 5.6 each of these is a Hermite polynomial applied to $d_{u,i}$, whose degree equals the multiplicity of the variable $d_{u,i}$. □

To better capture the combinatorics of the p_G, we look at matchings of the edge endpoints incident to a given vertex. More specifically we use a collection M of (partial or perfect) matchings on incident edges, one for each vertex.

Definition 5.11. *Let $\mathcal{PM}(G)$ be the set of all perfect matching collections of the edges incident to each vertex of G. Each element of $\mathcal{PM}(G)$ specifies $|V(G)|$ perfect matchings, and the perfect matching for vertex v is on* $\deg(v)$ *elements.*

Let $\mathcal{M}(G)$ denote the set of all partial or perfect matching collections of the edges incident to each vertex of G.

Definition 5.12. *For $M \in \mathcal{M}(\mathcal{G})$, define the routed graph* $\mathrm{route}(M)$ *to be the graph obtained by connecting up edge endpoints that are matched at each vertex v. Closed cycles are deleted, and paths are replaced by a single edge between the final path endpoints.*

Definition 5.13. *For $M \in \mathcal{M}(\mathcal{G})$, define* $\mathrm{cycles}(M)$ *to be the number of closed cycles formed by routing.*

We give an example in Fig. 5.2. The graph on the left has 5 vertices, and 10 edges denoted by solid lines. The edges are partially matched up at each vertex using the dashed edges. The right side shows the result of routing. $\mathrm{cycles}(M) = 1$ and one closed cycle, the triangle, was deleted.

With these definitions, we have the following topological definition of p_G.

Lemma 5.14 (Routing definition of p_G)**.**

$$p_G = \sum_{M\in\mathcal{M}(G)} m_{\mathrm{route}(M)} \cdot n^{\mathrm{cycles}(M)} \cdot (-1)^{|M|}.$$

A given graph K can appear as $\mathrm{route}(M)$ for several different matchings M (even with different numbers of cycles). This gives rise to interesting and nontrivial coefficients on the monomials m_K.

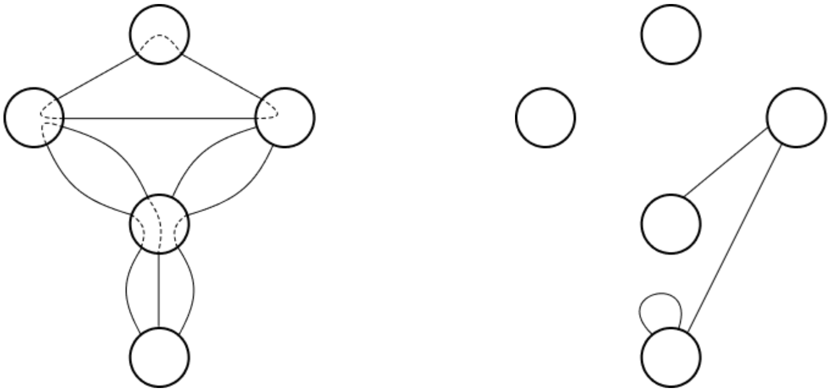


Figure 5.2: Left: Unrouted graph with dashed edges denoting the partial matching collection. Right: Result of routing.

Proof of Lemma 5.14. Expand the Wick product,

$$
\begin{aligned}
{:}m_G{:} &= \sum_{\sigma:E(G)\to[n]} : \left(\prod_{\{u,v\}\in E(G)} d_{u,i} d_{v,i} \right) : \\
&= \sum_{\sigma:E(G)\to[n]} \sum_{\substack{\text{matchings } M \text{ on} \\ \text{edge endpoints}}} (-1)^{|M|} \prod_{((u,e),(v,f))\in M} \mathbb{E}[d_{u,\sigma(e)} d_{v,\sigma(f)}] \prod_{\text{unmatched } (u,e)} d_{u,\sigma(e)} \\
&= \sum_{\substack{\text{matchings } M \text{ on} \\ \text{edge endpoints}}} (-1)^{|M|} \sum_{\sigma:E(G)\to[n]} \prod_{((u,e),(v,f))\in M} \mathbb{E}[d_{u,\sigma(e)} d_{v,\sigma(f)}] \prod_{\text{unmatched } (u,e)} d_{u,\sigma(e)} \,.
\end{aligned}
$$

The only way the expectation can be nonzero is if (1) $u = v$ for all matched pairs, and (2) $\sigma(e) = \sigma(f)$ for every two matched incident edges. Therefore, M routes $E(G)$ into paths and closed cycles, and σ must be constant on any path or cycle. For each cycle, there are n possible choices for σ, giving a factor of $n^{\text{cycles}(M)}$. For each path, summing over the n possible choices for σ gives $\langle d_s, d_t \rangle$ where s, t are the start and end vertices of the path. The $\langle d_s, d_t \rangle$ are collected into $m_{\text{route}(M)}$. □

As a consequence of the routing definition we have

Lemma 5.15. *The polynomials p_G are orthogonally invariant.*

Corollary 5.16. *p_G is equal to the degree-orthogonal Gram-Schmidt process on m_G.*

Proof. The Wick product is degree-orthogonal and monic. The previous lemma shows that the p_G are orthogonally invariant. Therefore they match the result of Gram-Schmidt by Fact 5.2. □

Example 5.17. *Let G be the graph with vertices $V(G) = \{u, v_1, v_2, v_3\}$ and edges $E(G) = \{\{u, v_1\}, \{u, v_2\}, \{u, v_3\}\}$. We have that*

$$\begin{aligned} m_G &= \langle d_u, d_{v_1} \rangle \langle d_u, d_{v_2} \rangle \langle d_u, d_{v_3} \rangle \\ &= \sum_{\text{distinct } i,j,k \in [n]} d_{u,i} d_{u,j} d_{u,k} d_{v_1,i} d_{v_2,j} d_{v_3,k} + \sum_{i \neq j \in [n]} d_{u,i}^2 d_{u,j} d_{v_1,i} d_{v_2,i} d_{v_3,j} \\ &+ \sum_{i \neq j \in [n]} d_{u,i}^2 d_{u,j} d_{v_1,i} d_{v_2,j} d_{v_3,i} + \sum_{i \neq j \in [n]} d_{u,i}^2 d_{u,j} d_{v_1,j} d_{v_2,i} d_{v_3,i} + \sum_{i \in [n]} d_{u,i}^3 d_{v_1,i} d_{v_2,i} d_{v_3,i}. \end{aligned}$$

Replacing the monomial $d_{u,i}^2$ with the corresponding Hermite polynomial $d_{u,i}^2 - 1$ and replacing the monomial $d_{u,i}^3$ with the corresponding Hermite polynomial $d_{u,i}^3 - 3d_{u,i}$, we have that

$$\begin{aligned} p_G = &\sum_{\text{distinct } i,j,k \in [n]} d_{u,i} d_{u,j} d_{u,k} d_{v_1,i} d_{v_2,j} d_{v_3,k} + \sum_{i \neq j \in [n]} (d_{u,i}^2 - 1) d_{u,j} d_{v_1,i} d_{v_2,i} d_{v_3,j} \\ &+ \sum_{i \neq j \in [n]} (d_{u,i}^2 - 1) d_{u,j} d_{v_1,i} d_{v_2,j} d_{v_3,i} + \sum_{i \neq j \in [n]} (d_{u,i}^2 - 1) d_{u,j} d_{v_1,j} d_{v_2,i} d_{v_3,i} \\ &+ \sum_{i \in [n]} (d_{u,i}^3 - 3d_{u,i}) d_{v_1,i} d_{v_2,i} d_{v_3,i} \end{aligned}$$

The term $-d_{u,j} d_{v_1,i} d_{v_2,i} d_{v_3,j}$ and one of the 3 terms $-d_{u,i} d_{v_1,i} d_{v_2,i} d_{v_3,i}$ correspond to the collection of matchings M where v_1 is matched to v_2 at u (and all other matchings are trivial). Summing these terms over all $i \neq j \in [n]$ gives $-\langle d_{v_1}, d_{v_2} \rangle \langle d_u, d_{v_3} \rangle$.

Similarly, the term $-d_{u,j} d_{v_1,i} d_{v_2,j} d_{v_3,i}$ and one of the 3 terms $-d_{u,i} d_{v_1,i} d_{v_2,i} d_{v_3,i}$ correspond to the collection of matchings M where v_1 is matched to v_3 at u (and all other matchings are trivial). Summing these terms over all $i \neq j \in [n]$ gives $-\langle d_{v_1}, d_{v_3} \rangle \langle d_u, d_{v_2} \rangle$.

Finally, the term $-d_{u,j} d_{v_1,j} d_{v_2,i} d_{v_3,i}$ and one of the 3 terms $-d_{u,i} d_{v_1,i} d_{v_2,i} d_{v_3,i}$ correspond to the collection of matchings M where v_2 is matched to v_3 at u (and all other matchings are trivial). Summing these terms over all $i \neq j \in [n]$ gives $-\langle d_{v_2}, d_{v_3} \rangle \langle d_u, d_{v_1} \rangle$.

Putting everything together,

$$p_G = \langle d_u, d_{v_1} \rangle \langle d_u, d_{v_2} \rangle \langle d_u, d_{v_3} \rangle - \langle d_{v_1}, d_{v_2} \rangle \langle d_u, d_{v_3} \rangle - \langle d_{v_1}, d_{v_3} \rangle \langle d_u, d_{v_2} \rangle - \langle d_{v_2}, d_{v_3} \rangle \langle d_u, d_{v_1} \rangle .$$

5.2.3 Formulas and properties

Note that the m_G are not completely linearly independent. For example, if $n = 1$, then m_G is determined by its degrees on each vertex. Despite this, the low-degree monomials are linearly independent.

Lemma 5.18. *The set of m_G for $|E(G)| \leq n$ is linearly independent.*

Proof. Suppose that $\sum_{G:|E(G)| \leq n} c_G m_G = 0$; we show $c_G = 0$. Each inner product $\langle d_u, d_v \rangle$ can be expanded as $\sum_{i=1}^{n} d_{u,i} d_{v,i}$. In this way, each edge gets a label from 1 to n. Expanding m_G,

$$m_G = \sum_{\sigma : E(G) \to [n]} \prod_{\{u,v\} \in E(G)} d_{u,\sigma(\{u,v\})} d_{v,\sigma(\{u,v\})}.$$

Since $|E(G)| \leq n$, one monomial that appears in m_G will have σ assign a distinct label to each edge. We claim that this monomial appears in the sum with coefficient c_G: because the edge labels are distinct, we can recover the graph G from the monomial. Therefore $c_G = 0$. □

For low-degree polynomials the orthogonalizations p_G will therefore be a basis.

In the language of matching collections, we have the following formula for the expectation $\mathbb{E}[m_G]$.

Lemma 5.19. $\mathbb{E}[m_G] = 0$ *if some vertex in G has odd degree. Otherwise,*

$$\mathbb{E}[m_G] = \sum_{M \in \mathcal{PM}(G)} n^{\text{cycles}(M)}.$$

Proof. Expanding m_G and grouping by vertex,

$$m_G = \sum_{\sigma : E \to [n]} \prod_{u \in V, i \in [n]} d_{u,i}^{\#\{e \ni u \,:\, \sigma(e) = i\}}.$$

Taking expectations, the $d_{u,i}$ are independent Gaussians. If one of the vertices has odd degree, one of the labels i will necessarily occur an odd number of times at that vertex and the overall expectation will be zero. Otherwise, $\mathbb{E}\left[Z^{2k}\right] = (2k-1)!!$ for $Z \sim \mathcal{N}(0,1)$. The expression $(2k-1)!!$ counts the number of perfect matchings of $2k$ elements; in this case when computing $\mathbb{E}\left[d_{u,i}^{\#\{e \ni u \,:\, \sigma(e) = i\}}\right]$ these should be thought of as summing 1 for each perfect matching of the edges incident to u which are labeled i. In summary, each σ sums over a subset of $\mathcal{PM}$.

Now fix a given collection of perfect matchings $M \in \mathcal{PM}$; which σ contribute to it? We require that, at each vertex, every pair of endpoints matched in M are assigned the same label. Therefore, in any cycle formed by route(M), the labeling σ must assign all edges of the cycle the same label. These labels can be any number from $[n]$, and disjoint cycles don't affect each other. Therefore there are $n^{\text{cycles}(M)}$ such σ. □

Corollary 5.20. *The magnitude of* $\mathbb{E}[m_G]$ *is* n^k *where k is the maximum number of cycles into which* $E(G)$ *can be partitioned (note that this is NP-hard to compute from G).*

To computationally compute $\mathbb{E}[m_G]$, an alternate recursive method is to use the combinatorially-flavored Isserlis' theorem (also known as Wick's lemma).

Lemma 5.21 (Isserlis' theorem)**.** *Fix vectors* $d_1, \ldots, d_{2k} \in \mathbb{R}^n$*. Then for v a standard n-*

dimensional Gaussian random variable,

$$\mathop{\mathbb{E}}_{v}[\langle v, d_1\rangle \cdots \langle v, d_{2k}\rangle] = \sum_{\substack{\textit{perfect matchings}\\ M \textit{ on } [2k]}} \prod_{(u,v)\in M} \langle d_u, d_v\rangle \,.$$

Observe also that the expectation is zero when there are an odd number of inner products.

Specifically, use the following minor generalization.

Lemma 5.22. *For fixed $d_1, \dots, d_{2k} \in \mathbb{R}^n$ and $v \sim \mathcal{N}(0, \mathrm{Id}_n)$,*

$$\mathop{\mathbb{E}}_{v}[\langle v, v\rangle^{2l} \langle v, d_1\rangle \cdots \langle v, d_{2k}\rangle] = n(n+2)\cdots(n+2l-2) \sum_{\substack{\textit{perfect matchings}\\ M \textit{ on } [2k]}} \prod_{(u,v)\in M} \langle d_u, d_v\rangle \,.$$

Proof. By expressing $v = \sqrt{Q}s$ where $Q \sim \chi^2(n)$ and $s \in_{\mathrm{R}} S^{n-1}$ are independent, this follows from the standard Isserlis theorem in a similar way as we will show in Lemma 5.40. □

The generalization can be iterated to compute $\mathbb{E}[m_G]$ for a given graph G. We take the expectation over the vectors d_u one at a time, and each application reduces our expression to a sum over graphs that no longer involve u.

The proof of Proposition 5.4 shows that p_G have a stronger "ultra-orthogonality" property. If G and H have different graph degree at u, then only taking the expectation over d_u already results in the zero polynomial.

Lemma 5.23. *Let G and H be two multigraphs. If $\deg_G(u) \neq \deg_H(u)$ for some $u \in V$, then*

$$\mathop{\mathbb{E}}_{d_u \sim \mathcal{N}(0,\mathrm{Id}_n)}[p_G \cdot p_H] = 0.$$

We now derive an explicit formula for the inner product and variance of p_G. For two graphs G, H on V, we define $G \cup H$ to be the disjoint union of the edges (the edge multiplicity in $G \cup H$ is the sum of the multiplicities in G and H).

Definition 5.24. *For two multigraphs, write $G \leftrightarrow H$ if $\deg_G(u) = \deg_H(u)$ for all $u \in V$.*

Definition 5.25. *Let $\mathcal{PM}(G, H) \subseteq \mathcal{PM}(G \cup H)$ be perfect matching collections such that at each vertex v, the matching goes between edges incident to v in G and edges incident to v in H. Note that if $G \not\leftrightarrow H$, then $\mathcal{PM}(G, H)$ is empty.*

Lemma 5.26. *Let G and H be two multigraphs.*

$$\mathbb{E}[p_G \cdot p_H] = \sum_{M \in \mathcal{PM}(G,H)} n^{\mathrm{cycles}(M)}$$

Remark 5.27. *Determining the maximum number of cycles in $M \in \mathcal{PM}(G, H)$ is NP-hard via a slight modification of [Hol81]. This remains true if we restrict the cycles to be simple.*

Proof. Use the routing definition of p_H,

$$p_G \cdot p_H = \sum_{\substack{M_1 \in \mathcal{M}(G),\\ M_2 \in \mathcal{M}(H)}} m_{\mathrm{route}_G(M_1)} \cdot m_{\mathrm{route}_H(M_2)} \cdot n^{\mathrm{cycles}_G(M_1)+\mathrm{cycles}_H(M_2)} \cdot (-1)^{|M_1|+|M_2|}.$$

Taking expectations, by Lemma 5.19 we sum over all perfect matchings of $\mathrm{route}_G(M_1) \cup \mathrm{route}_H(M_2)$ which "complete" the partial matchings M_1 and M_2, when viewed as a matching on the graph $G \cup H$. The power of n is the number of cycles in the completed matching. The net effect is to sum over all perfect matching collections in $G \cup H$,

$$\mathbb{E}[p_G p_H] = \sum_{M \in \mathcal{PM}(G \cup H)} n^{\mathrm{cycles}_{G\cup H}(M)} \sum_{\text{pick some } M\text{-matched pairs to be in } M_1 \text{ or } M_2} (-1)^{|M_1|+|M_2|}.$$

The inner summation often cancels to zero. In the graph $G \cup H$, each edge-vertex incidence comes from either G or H. We can only add an M-matched pair to M_1 if both matched edge-vertex incidences come from G; similarly only matched pairs which are both in H can be picked for M_2. If there are any such pairs, the inner summation is automatically zero.

The remaining terms are those M in which, at every vertex, the perfect matching is a perfect matching between incoming edges in G and those in H – that is, matching collections in $\mathcal{PM}(G, H)$. For these terms, the inner summation is trivially 1, which finishes the proof. □

Corollary 5.28. $n^{|E(G)|} \le \mathbb{E}[p_G^2] \le |E(G)|^{2|E(G)|} \cdot n^{|E(G)|}$.

Proof. Using the result of Lemma 5.26, we claim $\max_{M \in \mathcal{PM}(G,G)} \mathrm{cycles}(M) = |E(G)|$. On the one hand, $|E(G)|$ is achievable by matching each edge with its duplicate to create 2-cycles. On the other hand, every cycle in $\mathrm{route}(M)$ needs at least two edges (there can be no self-loops as matchings with self-loops are not in $\mathcal{PM}(G, G)$). This shows $n^{|E(G)|} \le \mathbb{E}\, p_G^2 \le |\mathcal{PM}(G, G)| \cdot n^{|E(G)|}$.

$\mathcal{PM}(G, G)$ consists of choosing a perfect matching at each vertex between two sets of size $\deg(v)$.

$$|\mathcal{PM}(G, G)| = \prod_{v \in V} \deg(v)! \le |E(G)|^{2|E(G)|}.$$

□

5.3 Polynomial Basis for the Spherical Setting

Let $S^{n-1} = \{x \in \mathbb{R}^n : \|x\|_2 = 1\}$. With the d_u drawn uniformly and independently from S^{n-1} instead of the Gaussian distribution, for each multigraph G with no self-loops (reflecting the fact that $\langle d_u, d_u \rangle = 1$) we construct a polynomial p_G. We again construct the polynomials in terms of routings (Definition 5.34) and study the inner product (Section 5.3.3) and variance (Corollary 5.46). For the most part, the proofs in this section mirror their counterparts in the previous section, with the notable exception of the inner product formula, which exhibits surprising difficulties.

5.3.1 Definitions of inner product polynomials

Let $\alpha \in \mathbb{N}^n$ be a multi-index and $|\alpha| := \sum_{i=1}^n \alpha_i$. We will need the Maxwell representation of harmonic polynomials [DX13, Theorem 1.1.9]. Concretely, let the spherical harmonic $s_\alpha : S^{n-1} \to \mathbb{R}$ be (the restriction to S^{n-1} of the function on $\mathbb{R}^n$)

$$s_\alpha(x) = \|x\|^{2|\alpha|+n-2} \frac{\partial}{\partial x^\alpha} \|x\|^{-n+2}$$

and then scaled to be monic. An alternate method to write down s_α is to first write down the Hermite polynomial $\prod_{i=1}^n h_{\alpha_i}(x_i)$ and then multiply each non-leading monomial of total degree $|\alpha| - 2k$ by approximately[4] n^{-k}. More precisely, we let $x^{\underline{\underline{k}}}$ be notation for the "fall-by-2" falling factorial,

$$x^{\underline{\underline{k}}} := x(x-2)(x-4)\cdots(x-2k+2),$$

and let $x^{\underline{\underline{-k}}} := 1/x^{\underline{\underline{k}}}$. We also define $x^{\overline{\overline{k}}}$ likewise for rise-by-2. Then:

Fact 5.29. *To form s_α from h_α, multiply monomials with degree $|\alpha| - 2k$ by $(n + 2|\alpha| - 4)^{\underline{\underline{-k}}}$.*

We will need the moments of the uniform distribution on the sphere (using the notation introduced above):

Fact 5.30.

$$\mathop{\mathbb{E}}_{x \in_{\mathrm{R}} S^{n-1}}[x^\alpha] = \begin{cases} n^{\overline{\overline{-|\alpha|/2}}} \cdot \mathbb{E}_{Z \sim \mathcal{N}(0, \mathrm{Id}_n)}[Z^\alpha] & \text{If } \alpha_i \text{ even for all } i \\ 0 & \text{Otherwise} \end{cases}$$

These spherical harmonics are degree-orthogonal (as functions of a single vector):

Fact 5.31. *If $|\alpha| \neq |\beta|$, then $\mathop{\mathbb{E}}_{x \in_{\mathrm{R}} S^{n-1}}[s_\alpha(x) s_\beta(x)] = 0$.*

We remark that $\{s_\alpha : \alpha \in \mathbb{N}^n\}$ is not completely linearly independent as functions on S^{n-1} because of the identity $\langle v, v\rangle = 1$:

Fact 5.32. *For each k, the set $\{s_\alpha : |\alpha| \leq k, \alpha_n = 0 \text{ or } 1\}$ is a basis for the set of degree-($\leq k$) polynomial functions on S^{n-1}. The same holds for the monomials x^α.*

[4] Multiplying the monomials by exactly n^{-k} creates polynomials orthogonal under the distribution $\mathcal{N}(0, \mathrm{Id}_n/n)$, which is similar to the unit sphere.

We now give three definitions of the orthogonal polynomials for the spherical case which are analogous to Definitions 5.10, 5.14, and 5.9 for the Gaussian case:

Definition 5.33 (Spherical harmonic sum definition)**.** *Define p_G by*

$$p_G = \sum_{\sigma : E(G) \to [n]} \prod_{u \in V} s_{\textit{histogram of } \{\sigma(e) : e \ni u\}}(d_u).$$

Definition 5.34 (Routing definition)**.** *Define p_G by*

$$p_G = \sum_{M \in \mathcal{M}(G)} m_{\text{route}(M)} \cdot n^{\text{cycles}(M)} \cdot (-1)^{|M|} \cdot \prod_{v \in V} (n + 2\deg(v) - 4)^{\underline{\underline{-|M_v|}}}$$

where M_v is the partial matching of incident edges at v.

Definition 5.35 (Generic construction from Proposition 5.4)**.** *To construct p_G, expand the function m_G in the basis of spherical harmonics as a function of $d_{u,i}$ then truncate to the top-level coefficients of degree $2|E(G)|$.*

Lemma 5.36. *The three definitions above are equivalent.*

Proof. Definitions 5.33 and 5.35 agree once we check that the leading monomial in Definition 5.33 is m_G.

Definitions 5.33 and 5.34 agree as a consequence of equality between Lemma 5.10 and Lemma 5.14 in the Gaussian case by the following argument. For reference, we recall the two equal formulas for p_G in the Gaussian case,

$$p_G = \sum_{\sigma : E(G) \to [n]} \prod_{u \in V, i \in [n]} h_{|\{e \ni u \,:\, \sigma(e) = i\}|}(d_{u,i}) \tag{5.1}$$

$$p_G = \sum_{M \in \mathcal{M}(G)} m_{\text{route}(M)} \cdot n^{\text{cycles}(M)} \cdot (-1)^{|M|}. \tag{5.2}$$

For each fixed $\delta \in V^{\mathbb{N}}$, let us restrict to only the monomials in the variables $d_{u,i}$ with total degree $\delta(u)$ on the variables $\{d_{u,i} : i \in [n]\}$. We clearly still have equality between Eq. (5.1) and Eq. (5.2) after making this restriction. The equality still holds if we multiply both sides by an appropriate function of n; we choose this function of n to be the product that appears on the right side of Definition 5.34, which only depends on δ. This clearly converts Eq. (5.2) into Definition 5.34. Due to the choice of function, it also turns Eq. (5.1) into Definition 5.33 because of the conversion between h_α and s_α in Fact 5.29. □

As a consequence of the routing definition, we have:

Lemma 5.37. *The polynomials p_G are orthogonally invariant.*

Corollary 5.38. *Definitions 5.33, 5.34, 5.35 are equal to the output of the degree-orthogonal Gram-Schmidt process on m_G.*

5.3.2 Formulas and properties

The monomials m_G are not completely linearly independent as functions on $(S^{n-1})^V$. We restrict ourselves to the set of low-degree functions, which are linearly independent.

Lemma 5.39. *The set of m_G with $|E(G)| \leq n-1$ is linearly independent as functions on $(S^{n-1})^V$.*

Proof. Suppose $\sum_{G:|E(G)|\leq n} c_G m_G = 0$ where c_G are not all zero, and let G be a nonzero graph with maximum number of edges. Expanding m_G,

$$m_G = \sum_{\sigma:E(G)\to[n]} \prod_{\{u,v\}\in E(G)} d_{u,\sigma(\{u,v\})} d_{v,\sigma(\{u,v\})}.$$

Letting σ be an injective assignment of labels from $[n-1]$ (which exists because $|E(G)| \leq n-1$), we claim that the corresponding monomial, which we call the "special monomial", is uncancelled and appears with coefficient c_G.

First, the relations $\langle d_u, d_u\rangle = 1$ mean that polynomials do not have a unique representation as functions on $(S^{n-1})^V$. We amend this by using the relations to reduce the degree of variable $d_{u,n}$ to 0 or 1 for each vertex u, replacing $d_{u,n}^2 = 1 - \sum_{i=1}^{n-1} d_{u,i}^2$. Nothing needs to be done for the special monomial.

After performing the replacement, the special monomial still does not arise from any other graphs. This is because the reduction step must either lower the degree, or introduce a variable with degree 2, whereas the special monomial is multilinear and was chosen to have maximum degree. Therefore, the special monomial has coefficient c_G, which is nonzero, a contradiction. □

There is a spherical Isserlis theorem which gives a recursive method to compute $\mathbb{E}[m_G]$.

Lemma 5.40 (Spherical Isserlis theorem)**.** *Fix vectors $d_1, \dots, d_{2k} \in \mathbb{R}^n$. Then for $v \in_R S^{n-1}$,*

$$\mathop{\mathbb{E}}_{v}[\langle v, d_1\rangle \cdots \langle v, d_{2k}\rangle] = n^{\overline{\overline{-k}}} \sum_{\substack{\textit{perfect matchings}\\ \mathcal{M} \textit{ on } [2k]}} \prod_{(u,v)\in\mathcal{M}} \langle d_u, d_v\rangle .$$

Observe also that the expectation is zero when there are an odd number of inner products.

Proof. This follows from the standard Isserlis theorem. Let $Q \sim \chi^2(n)$ be a chi-square random variable with n degrees of freedom, independent from v. Then

$$\mathop{\mathbb{E}}_{v,Q}\left[\left\langle \sqrt{Q}v, d_1\right\rangle \cdots \left\langle \sqrt{Q}v, d_{2k}\right\rangle\right] = \mathop{\mathbb{E}}_{Z\sim\mathcal{N}(0,\mathrm{Id}_n)}[\langle Z, d_1\rangle \cdots \langle Z, d_{2k}\rangle].$$

Factoring out Q^k, the left-hand side is

$$\mathop{\mathbb{E}}_{v,Q}\left[\left\langle \sqrt{Q}v, d_1\right\rangle \cdots \left\langle \sqrt{Q}v, d_{2k}\right\rangle\right] = \mathop{\mathbb{E}}_{Q}[Q^k] \cdot \mathop{\mathbb{E}}_{v}[\langle v, d_1\rangle \cdots \langle v, d_{2k}\rangle] = n^{\overline{\overline{k}}} \mathop{\mathbb{E}}_{v}[\langle v, d_1\rangle \cdots \langle v, d_{2k}\rangle].$$

By the Gaussian Isserlis theorem, the right-hand side equals

$$\mathop{\mathbb{E}}_{Z \sim \mathcal{N}(0, \mathrm{Id}_n)}[\langle Z, d_1\rangle \cdots \langle Z, d_{2k}\rangle] = \sum_{\substack{\text{perfect matchings} \\ \mathcal{M} \text{ on } [2k]}} \prod_{(u,v)\in\mathcal{M}} \langle d_u, d_v\rangle .$$

Dividing by $n^{\overline{\overline{k}}}$ proves the claim. □

We also have explicit formulas based on matching collections,

Lemma 5.41. $\mathbb{E}[m_G] = 0$ *if there is a vertex of odd degree, otherwise,*

$$\mathbb{E}[m_G] = \prod_{v \in V} n^{\overline{\overline{-\deg(v)/2}}} \sum_{M \in \mathcal{PM}(G)} n^{\mathrm{cycles}(M)}.$$

Proof. The proof goes through exactly as in the Gaussian case, but plug in the spherical moments which contribute the rising factorial terms. □

Remark 5.42. *Lemma 5.41 is still valid if G has self-loops.*

The "ultra orthogonality" property follows from Proposition 5.4.

Lemma 5.43. *Let G and H be two multigraphs. If* $\deg_G(u) \neq \deg_H(u)$ *for some* $u \in V$*, then* $\mathbb{E}_{d_u \in_R S^{n-1}}[p_G \cdot p_H] = 0$.

5.3.3 Inner product

Unfortunately, we do not have a clean formula for the inner product of two spherical polynomials. Compared to the Gaussian case, there are several complications. First, in the spherical case some polynomials with $G \leftrightarrow H$ are orthogonal (whereas in the Gaussian case p_G and p_H are orthogonal iff $G \not\leftrightarrow H$, via Lemma 5.26). For example, the following two polynomials are orthogonal:

$$p_G = x_{12}x_{13}x_{45} - \frac{x_{23}x_{45}}{n}, \qquad p_H = x_{14}x_{15}x_{23} - \frac{x_{45}x_{23}}{n}.$$

This shows that extra cancellations occur in the spherical case. Second, when n is small some of the p_G are degenerate. For example, if G is a triangle and $n = 2$ then $p_G = 0$. Third, even in the asymptotic regime of large n and constant-size graphs, in the spherical case the inner product may be negative (whereas in the Gaussian case the inner product is always non-negative). For example, this occurs if $E(G), E(H)$ partition the edges of K_5, with the inner 5-cycle in G and the outer 5-cycle in H, as in Fig. 5.3.

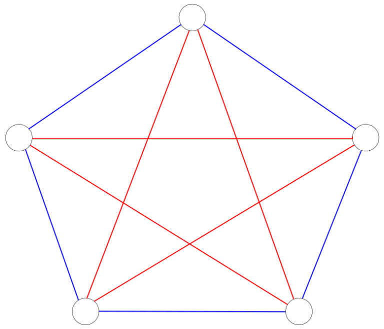


Figure 5.3

In this case it can be computed (see [JP22, Appendix C]) that

$$\mathbb{E}[p_G \cdot p_H] = \frac{-8(n-1)(n-2)(n-4)}{n^8(n+2)^4}.$$

Interestingly, we conjecture that negative inner product can only occur if the graph $G \cup H$ is nonplanar.

To attack these complications, we first give a general expression for the inner product. We use it to upper bound the magnitude of the inner product, showing that it's no larger than the Gaussian case, up to normalization (Corollary 5.47). We then study some situations when cancellations occur in an effort to determine the exact magnitude in n of the inner product. Due to the inherent difficulties, this section is a bit technical.

The proof strategy we use is to consider the contribution c_M from each matching collection $M \in \mathcal{PM}(G \cup H)$ and then isolate cancellations that occur between these terms (similarly to how the inner product was computed in the Gaussian case, Lemma 5.26).

Definition 5.44. *For some $M \in \mathcal{PM}(G \cup H)$, define a G-pair as a pair of matched endpoints in M where both come from G. An H-pair and a (G, H)-pair are defined analogously.*

Let $g_M(v)$ denote the number of G-pairs at vertex v and g_M denote the total number of G-pairs.

If $G \not\leftrightarrow H$ then $\mathbb{E}[p_G p_H] = 0$ by Lemma 5.43, so we may assume $G \leftrightarrow H$.

Lemma 5.45. *Let G, H be arbitrary multigraphs such that $G \leftrightarrow H$. Let $d(v) = \deg_G(v) = \deg_H(v)$. Then*

$$\mathbb{E}[p_G \cdot p_H] = \prod_{v \in V} n^{\overline{\overline{-d(v)}}} \sum_{M \in \mathcal{PM}(G \cup H)} c_M$$

where the coefficients c_M are

$$c_M = n^{\text{cycles}(M)} \prod_{v \in V} \frac{(-2)^{\underline{\underline{g_M(v)}}}}{(n + 2d(v) - 4)^{\underline{\underline{g_M(v)}}}}.$$

Proof. By orthogonality, $\mathbb{E}[p_G \cdot p_H] = \mathbb{E}[p_G \cdot m_H]$. Using the routing definition,

$$p_G \cdot m_H = \sum_{M \in \mathcal{M}(G)} m_{\mathrm{route}_G(M)} \cdot m_H \cdot n^{\mathrm{cycles}_G(M)} \cdot (-1)^{|M|} \cdot \prod_{v \in V} (n + 2d(v) - 4)^{\underline{\underline{-|M_v|}}}.$$

Taking expectations[5] using Lemma 5.41, we expand $\mathbb{E}[m_{\mathrm{route}_G(M)} \cdot m_H]$ into a sum over all completions C of the partial matching M (on the graph $G \cup H$). As in the Gaussian case, we collect terms based on the overall matching $M \cup C \in \mathcal{PM}(G \cup H)$. Performing the grouping of terms, we have

$$\begin{aligned}
\mathbb{E}\, p_G m_H &= \sum_{M \in \mathcal{PM}(G \cup H)} n^{\mathrm{cycles}_{G \cup H}(M)} \sum_{S \subseteq G\text{-pairs}} (-1)^{|S|} \cdot \prod_{v \in V} (n + 2d(v) - 4)^{\underline{\underline{-|S_v|}}} \cdot n^{\overline{\overline{-d(v)+|S_v|}}} \\
&= \sum_{M \in \mathcal{PM}(G \cup H)} n^{\mathrm{cycles}(M)} \prod_{v \in V} \sum_{S_v \subseteq G\text{-pairs at } v} (-1)^{|S_v|} (n + 2d(v) - 4)^{\underline{\underline{-|S_v|}}} \cdot n^{\overline{\overline{-d(v)+|S_v|}}} \\
&= \sum_{M \in \mathcal{PM}(G \cup H)} n^{\mathrm{cycles}(M)} \prod_{v \in V} \sum_{k=0}^{g_M(v)} \binom{g_M(v)}{k} (-1)^k (n + 2d(v) - 4)^{\underline{\underline{-k}}} \cdot n^{\overline{\overline{-d(v)+k}}}.
\end{aligned}$$

The inner summation (with v fixed) is

$$\begin{aligned}
&\sum_{k=0}^{g_M(v)} \binom{g_M(v)}{k} (-1)^k (n + 2d(v) - 4)^{\underline{\underline{-k}}} \cdot n^{\overline{\overline{-d(v)+k}}} \\
&= n^{\overline{\overline{-d(v)}}} \sum_{k=0}^{g_M(v)} \binom{g_M(v)}{k} (-1)^k (n + 2d(v) - 4)^{\underline{\underline{-k}}} \cdot (n + 2d(v) - 2)^{\underline{\underline{k}}} \\
&= \frac{n^{\overline{\overline{-d(v)}}}}{(n + 2d(v) - 4)^{\underline{\underline{g_M(v)}}}} \sum_{k=0}^{g_M(v)} \binom{g_M(v)}{k} (-1)^k (n + 2d(v) - 2g_M(v) - 2)^{\overline{\overline{g_M(v)-k}}} \cdot (n + 2d(v) - 2)^{\underline{\underline{k}}} \\
&= \frac{n^{\overline{\overline{-d(v)}}}}{(n + 2d(v) - 4)^{\underline{\underline{g_M(v)}}}} \sum_{k=0}^{g_M(v)} \binom{g_M(v)}{k} (n + 2d(v) - 2g_M(v) - 2)^{\overline{\overline{g_M(v)-k}}} \cdot (-n - 2d(v) + 2)^{\overline{\overline{k}}}.
\end{aligned}$$

[5] We should not remove the self-loops in $\mathrm{route}_G(M)$ which is permitted by Remark 5.42.

Using the umbral formula $(x+y)^{\overline{\overline{m}}} = \sum_{k=0}^{m} \binom{m}{k} x^{\overline{\overline{k}}} y^{\overline{\overline{m-k}}}$ [Rom05],

$$= \frac{n^{\overline{\overline{-d(v)}}}}{(n+2d(v)-4)^{\underline{\underline{g_M(v)}}}} (-2g_M(v))^{\overline{\overline{g_M(v)}}}$$
$$= \frac{n^{\overline{\overline{-d(v)}}}}{(n+2d(v)-4)^{\underline{\underline{g_M(v)}}}} (-2)^{\underline{\underline{g_M(v)}}}.$$

□

Corollary 5.46. *For G such that $|E(G)| \leq o(\log n / \log\log n)$,*

$$\mathbb{E}[p_G^2] = n^{-|E(G)|+o(1)}.$$

Proof. We have

$$\prod_{v \in V} n^{\overline{\overline{-d(v)}}} = \prod_{v \in V} n^{-d(v)+o(1)} = n^{-2|E(G)|+o(1)}.$$

The magnitude of the coefficient c_M is $n^{\text{cycles}(M)-g_M}$. Since G has no self-loops, the max number of cycles for $M \in \mathcal{PM}(G \cup G)$ is $|E(G)|$, therefore M has the largest magnitude of n if and only if M pairs each edge with a parallel edge from the other copy of the graph. For these M, $c_M = n^{|E(G)|}$. There is at least one such M and possibly up to $|PM(G,H)|$. Under the size assumption on G, $|PM(G,H)| = n^{o(1)}$ and therefore non-dominant terms are negligible,

$$\mathbb{E}[p_G^2] = n^{-2|E(G)|+|E(G)|+o(1)} = n^{-|E(G)|+o(1)}.$$

□

Up to the normalization factor of $\prod_{v \in V} n^{\overline{\overline{-d(v)}}}$, the inner product is bounded by the same formula from the Gaussian case.

Corollary 5.47. *Let G and H be two multigraphs such that $G \leftrightarrow H$ with degrees $d(v)$, and $|E(G)|, |E(H)| \leq o(\log n / \log\log n)$. Then*

$$|\mathbb{E}[p_G \cdot p_H]| \leq \prod_{v \in V} n^{\overline{\overline{-d(v)}}} \sum_{M \in \mathcal{PM}(G,H)} n^{\text{cycles}(M)+o(1)}.$$

Proof. From Lemma 5.45,

$$\mathbb{E}[p_G \cdot p_H] = \prod_{v \in V} n^{\overline{\overline{-d(v)}}} \sum_{M \in \mathcal{PM}(G \cup H)} c_M$$

$$= \prod_{v \in V} n^{\overline{\overline{-d(v)}}} \sum_{M \in \mathcal{PM}(G \cup H)} n^{\text{cycles}(M)} \prod_{v \in V} \frac{(-2)^{\underline{\underline{g_M(v)}}}}{(n + 2d(v) - 4)^{\underline{\underline{g_M(v)}}}}.$$

If M has both a G-pair and an H-pair at v, observe how the magnitude of c_M changes if we re-match them into two (G, H)-pairs to get a new matching M'. $g_M(v)$ goes down by 1. cycles(M) may increase by 1, decrease by 1, or stay the same. Therefore the magnitude of $c_{M'}$ is at least as large as c_M. Iterating this, the dominant terms are $M \in \mathcal{PM}(G, H)$, and the size assumption means they are dominant up to a $n^{o(1)}$ factor. □

There are often significantly more cancellations than the Gaussian case. We conjecture that the magnitude for planar graphs $G \cup H$ is given by the *simple* matchings $M \in \mathcal{PM}(G, H)$.

Definition 5.48. *For a multigraph G and $M \in \mathcal{PM}(G)$, we say that M is v-simple if v is visited at most once in each cycle induced by M. We say that M is simple if every cycle is simple.*

Conjecture 5.49. *Let G and H be two loopless multigraphs such that $G \cup H$ is planar, and $|E(G)|, |E(H)| \leq o(\log n / \log \log n)$. Then*

$$\mathbb{E}[p_G \cdot p_H] = \frac{1}{n^{|E(G)|+|E(H)|}} \cdot \left(\sum_{\textit{simple } M \in \mathcal{PM}(G,H)} n^{\text{cycles}(M)} \right) \cdot (1 \pm o(1)).$$

If there are no simple $M \in \mathcal{PM}(G, H)$, then the expectation is zero.

K_5 is a counterexample to an extension of the conjecture to non-planar graphs. $\mathbb{E}[p_G p_H] < 0$ for G and H equal to two 5-cycles despite that:

Proposition 5.50. *Decomposing K_5 into two 5-cycles G and H, there is no simple $M \in \mathcal{PM}(G, H)$.*

Proof. The cycles created by $M \in \mathcal{PM}(G, H)$ are necessarily even-length since they alternate between G and H edges. They can't be length-2 since $E(G) \cap E(H) = \emptyset$. Therefore there must be two cycles of lengths 4 and 6, or one cycle of length 10, but a length-6 or 10 cycle is not simple. □

There are other examples with K_5 minors with negative inner product. Taking G, H to be the red and blue edges in Fig. 5.4,

$$\mathbb{E}[p_G p_H] = \frac{-16(n-1)(n-2)(n-4)}{n^{11}(n+2)^5}.$$

Taking G, H to be the red and blue edges in Fig. 5.5,

$$\mathbb{E}[p_G p_H] = \frac{-16(n-1)(n-2)^2(n-4)}{n^{11}(n+2)^6}.$$

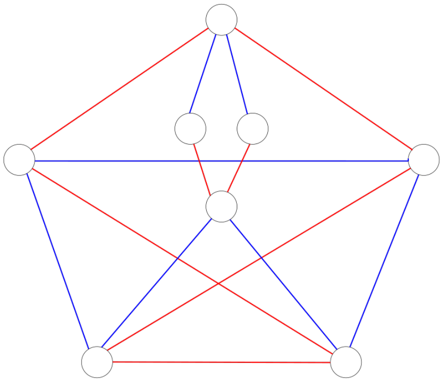


Figure 5.4

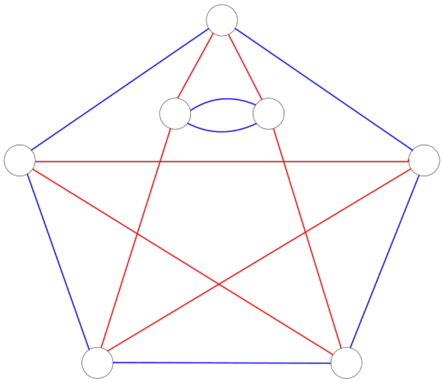


Figure 5.5

We don't know of any similar examples based on $K_{3,3}$. To lend support to the conjecture, we consider an approach that almost works, and show some intuition for why the failure of the approach is related to planarity (or is at least topological in nature).

Observe that fixing a matching collection off of a vertex v induces a matching of the edges incident to v. To wit, if you leave v along edge e, and follow the fixed matching around outside of v, you will eventually return to v along some edge. If the induced matching at v has a G-pair then we claim that summing over matchings at v produces zero.

Lemma 5.51. *Let M' be a perfect matching collection for all vertices except v. If there is a G-pair in the induced matching at v, then*

$$\sum_{M : M \text{ extends } M' \text{ at } v} c_M = 0.$$

Proof. Abbreviate the sum in the statement as $\sum_{M \succeq M'}$. The c_M factor out a term for vertices that are not v,

$$\sum_{M \succeq M'} c_M = \prod_{\substack{w \in V, \\ w \neq v}} \frac{(-2)^{\underline{\underline{g_{M'}(w)}}}}{(n+2d(w)-4)^{\underline{\underline{g_M(w)}}}} \sum_{M \succeq M'} n^{\text{cycles}(M)} \frac{(-2)^{\underline{\underline{g_M(v)}}}}{(n+2d(v)-4)^{\underline{\underline{g_M(v)}}}}.$$

We will argue that the latter sum is zero.

Let e_1 and e_2 be two edges which form a G-pair induced by the matchings M' outside of v. Consider the following map from matchings of the edges incident to v where e_1 is not matched to e_2 to matchings where e_1 is matched to e_2. If e_1 is matched to e_i and e_2 is matched to e_j then we match e_1 and e_2 and match e_i and e_j.

To invert this mapping, given a matching where e_1 is matched with e_2, we need to know which of the $d(v) - 1$ other matched pairs e_i and e_j to swap with and we need to know whether to match e_1 with e_i and e_2 with e_j or e_1 with e_j and e_2 with e_i.

Consider a given matching where e_1 is matched with e_2. Letting $c + 1$ be the number of cycles in the matching and $k + 1$ be the number of G-pairs at v, this matching gives a value of

$$n^{c+1} \cdot \frac{(-2)^{\underline{\underline{k}}}}{(n + 2d(v) - 4)^{\underline{\underline{k}}}} \cdot \frac{-2(k+1)}{n + 2d(v) - 2k - 4}.$$

We now show that this term cancels with the terms for all of the matchings where e_1 is not matched to e_2 which map to this matching. Observe an important property of the induced matching: for all re-matchings of e_1 and e_2 with e_i, e_j, the number of cycles decreases by exactly 1. For the $2(k+1)$ matchings where e_1 and e_2 are mixed with an H-pair, each such matching gives a value of

$$n^{c} \cdot \frac{(-2)^{\underline{\underline{k}}}}{(n + 2d(v) - 4)^{\underline{\underline{k}}}}.$$

For the $2d(v) - 2k - 4$ matchings where e_1 and e_2 are mixed with another G-pair or are matched with a (G, H)-pair, each such matching gives a value of

$$n^{c} \cdot \frac{(-2)^{\underline{\underline{k}}}}{(n + 2d(v) - 4)^{\underline{\underline{k}}}} \cdot \frac{-2(k+1)}{n + 2d(v) - 2k - 4}.$$

Adding these terms together and dividing by $n^c \cdot \frac{(-2)^{\underline{\underline{k}}}}{(n+2d(v)-4)^{\underline{\underline{k}}}}$, we obtain

$$\frac{-2(k+1)n}{n + 2d(v) - 2k - 4} + 2(k+1) - \frac{2(k+1)(2d(v) - 2k - 4)}{n + 2d(v) - 2k - 4} = 0.$$

□

The next corollary explains some cancellations.

Corollary 5.52. *If $G \cup H$ has a cut vertex v such that a component C of $(G \cup H) \setminus v$ has an unequal number of G edges and H edges incident to v, then $\mathbb{E}[p_G p_H] = 0$.*

Proof. Fixing any perfect matching collection on C, this necessarily induces either an H-pair or a G-pair at v. By the previous lemma, summing over the matchings at v yields zero. □

Even when we cannot apply Corollary 5.52, Lemma 5.51 can still be very useful in computing $\mathbb{E}[p_G p_H]$.

Example 5.53. *Consider the graphs G and H depicted in Fig. 5.6 where $V(G) = V(H) = \{1, 2, 3, 4\}$, $E(G) = \{\{1, 2\}, \{1, 2\}, \{3, 4\}, \{3, 4\}\}$, and $E(H) = \{\{2, 3\}, \{2, 3\}, \{4, 1\}, \{4, 1\}\}$.*

We can compute $\mathbb{E}[p_G p_H]$ as follows. Consider vertex 1 and the edges incident to it. We partition the collections of matchings based on how these edges are connected to each other in the remainder of the graph. We then sum over the possible matchings at vertex 1.

Let e_1 and e_2 be the two copies of $\{1,2\}$ and let e_3 and e_4 be the two copies of $\{1,4\}$.

1. *If there is a path from e_1 to e_2 and a path from e_3 to e_4 (outside of vertex 1) then by Lemma 5.51, everything cancels at vertex 1.*

2. *If there is a path from e_1 to e_3 and a path from e_2 to e_4 (outside of vertex 1) then summing over the matchings at vertex 1 gives*

$$\frac{n^2}{n(n+2)} + \frac{n}{n(n+2)} - \frac{2n}{n^2(n+2)} = \frac{n^2+n-2}{n(n+2)} = \frac{n-1}{n}$$

   To see this, note that the first term corresponds to the matching $\{e_1, e_3\}, \{e_2, e_4\}$ at vertex 1 as this gives two cycles and gives a factor of $\frac{1}{n(n+2)}$ for vertex 1. The second term corresponds to the matching $\{e_1, e_4\}, \{e_2, e_3\}$ at vertex 1 as this gives one cycle and gives a factor of $\frac{1}{n(n+2)}$ for vertex 1. The third term corresponds to the matching $\{e_1, e_2\}, \{e_3, e_4\}$ at vertex 1 as this gives one cycle and gives a factor of $\frac{-2}{n^2(n+2)}$ for vertex 1.

   Note that in order to have these paths, there must be G-H matchings at the other 3 vertices and there are 4 ways to do this and route e_1 to e_3 and e_2 to e_4. When there are G-H matchings at the other 3 vertices, each of these vertices gives a factor of $\frac{1}{n(n+2)}$

3. *The case when there is a path from e_1 to e_4 and a path from e_2 to e_3 behaves in the same way as the previous case.*

Adding everything together,

$$\mathbb{E}[p_G p_H] = 2 \cdot 4 \cdot \left(\frac{1}{n(n+2)}\right)^3 \cdot \frac{n-1}{n} = \frac{8(n-1)}{n^4(n+2)^3}$$

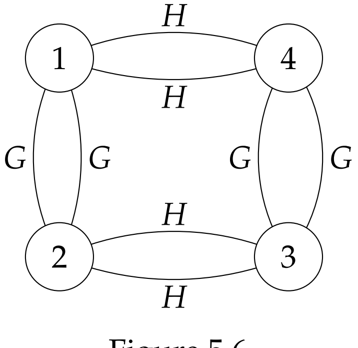


Figure 5.6

Lemma 5.51 suggests an approach for cancelling matchings $M \in \mathcal{PM}(G \cup H) \setminus \mathcal{PM}(G, H)$ or which are not simple. For a matching $M \in \mathcal{PM}(G \cup H)$ and a vertex v, define the *induced re-matching at v* to be $M' \in \mathcal{PM}(G \cup H)$ which agrees with M for all vertices except v, where it is the induced matching at v. Now, if there is a G-pair at v in M', then c_M can be cancelled out by summing over matchings at v and using Lemma 5.51. If M' has only (G, H)-pairs at v, observe that (1) the re-matching operation increases the magnitude of c_M, i.e. $c_{M'} \geq c_M$ and (2) the re-matching is guaranteed to be v-simple as well. Therefore c_M is not dominant, and it can be upper bounded by the v-simple (G, H)-matching M'. We would like to continue to the next vertex until the only remaining dominant matchings are between G and H and are simple. Based on this argument, we initially conjectured that Conjecture 5.49 held for all G, H not necessarily planar.

Unfortunately this strategy doesn't work as some matchings M may be needed to cancel out re-matchings for multiple distinct vertices. These matchings may appear with nonzero coefficients. The observation is that such matchings must have certain “crossing" structure. To explain this we specialize to the case where G, H have max degree 2. In this case

$$c_M = n^{\text{cycles}(M)} \prod_{v \in V} \begin{cases} 1 & d(v) \leq 1 \\ 1 & d(v) = 2 \text{ and no } G\text{-pair at } v \\ \frac{-2}{n} & d(v) = 2 \text{ and } G\text{-pair at } v \end{cases}$$

We group the c_M based on an overall matching $M \in \mathcal{PM}(G, H)$. This can be seen as applying the cancellation trick in Lemma 5.51 simultaneously to all vertices.

Definition 5.54. *For multigraphs G, H let V_4 be the set of vertices with $\deg_G(v) = \deg_H(v) = 2$. For $M \in \mathcal{PM}(G, H)$ and $v \in V_4$ let $M^{\oplus v} \in \mathcal{PM}(G \cup H)$ be defined by re-matching v into a G-pair and H-pair instead of two (G, H)-pairs. For $S \subseteq V_4$ let $M^{\oplus S}$ re-match all vertices in S.*[6]

Lemma 5.55. *Let G, H with max degree 2 and let V_4 be the set of degree-4 vertices in $G \cup H$.*

$$\mathbb{E}[p_G p_H] = \prod_{v \in V} n^{\overline{\overline{-d(v)}}} \sum_{M \in \mathcal{PM}(G,H)} n^{\text{cycles}(M)} \left(\sum_{S \subseteq V_4} (-1)^{|S|} \frac{n^{\text{cycles}(M^{\oplus S})}}{n^{\text{cycles}(M)+|S|}} \right).$$

Proof. Recall from Lemma 5.45,

$$\mathbb{E}[p_G p_H] = \prod_{v \in V} n^{\overline{\overline{-d(v)}}} \sum_{M \in \mathcal{PM}(G \cup H)} c_M.$$

Each $M \in \mathcal{PM}(G \cup H) \setminus \mathcal{PM}(G, H)$ has a coefficient of $c_M = n^{\text{cycles}(M)} \left(\frac{-2}{n}\right)^{g_M}$. For each G-pair, say at v, there are two ways to rematch at v to get two (G, H)-pairs. Split the coefficient $\frac{-2}{n}$ between these two rematchings. □

We say that a matching M is “uncancelled” if the inner summation over V_4 is nonzero.

[6] Note that the re-matchings in this definition are not necessarily induced re-matchings.

Let us fix G, H and an uncancelled term $M \in \mathcal{PM}(G, H)$ and assume for the sake of exposition that the inner summation is uncancelled and of order $\Omega(1)$; this is the maximum possible magnitude for the inner summation, as the next lemma shows.

Lemma 5.56. $\text{cycles}(M^{\oplus S}) \leq \text{cycles}(M) + |S|$

Proof. We have $\text{cycles}(M^{\oplus v}) \leq \text{cycles}(M) + 1$ since the only way to increase the number of cycles is if one cycle splits into two. The claim follows by induction. □

We show how the non-cancelling property in this case is due to topological properties of G and H (Lemma 5.61).

Definition 5.57. *$S \subseteq V_4$ is dominant if* $\text{cycles}(M^{\oplus S}) = \text{cycles}(M) + |S|$.

Definition 5.58. *For each cycle C in M, let $\text{gg}(C)$ be the set of vertices v such that*

(i) v is visited twice in C,

(ii) restricting the matching collection M to all vertices except v, the induced matching at v has a G-pair.

Let $\text{gg}(M) = \bigcup_{C \in M} \text{gg}(C)$.

In other words, $M^{\oplus v}$ for vertices in $\text{gg}(C)$ will split C into two subcycles. Fig. 5.7 gives an example with $G = \{\{1,2\},\{1,2\},\{2,3\},\{3,4\}\}, H = \{\{2,3\},\{3,4\}\}$. The same cycle C is drawn in two different ways. In the left image, the dashed lines are the matching collections for each vertex. In the right image, a vertex which is visited more than once by C is drawn more than once. The two vertices that are visited more than once are 2 and 3, and only 2 has an induced G-pair, so $\text{gg}(C) = \{2\}$.

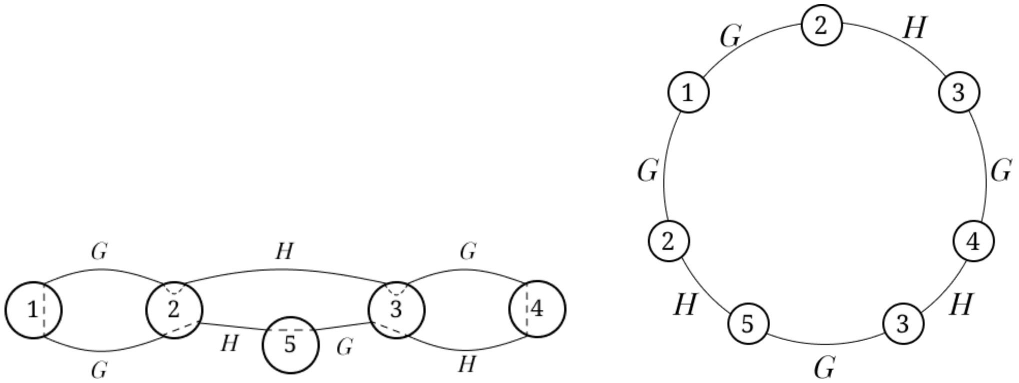


Figure 5.7: $\text{gg}(C) = \{2\}$.

Remark 5.59. *If one defines* $\text{hh}(M)$ *analogously using H, then* $\text{hh}(M) = \text{gg}(M)$.

Definition 5.60. *$S \subseteq V_4$ is non-crossing if for each cycle C, S is a non-crossing subset of C drawn in a circle.*

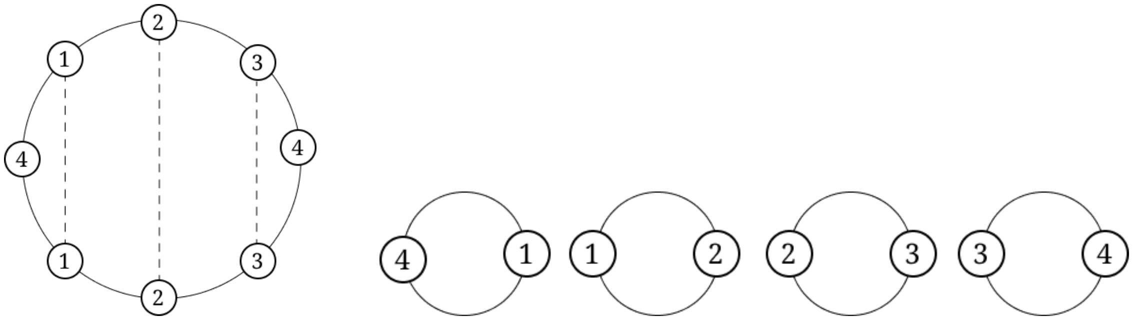


Figure 5.8: In the left circle, $\{1,2,3\}$ is a non-crossing subset of C. The right four circles show the result of the induced rematching of $\{1,2,3\}$.

In other words, the induced re-matching of all vertices in a non-crossing set S in a cycle will subdivide the cycle. A picture is given in Fig. 5.8.

The key lemma is that the dominant terms are precisely non-crossing subsets of $\mathrm{gg}(M)$:

Lemma 5.61. *S is dominant if and only if S is a non-crossing subset of* $\mathrm{gg}(M)$.

Proposition 5.62. *If S is dominant then any subset $S' \subseteq S$ is also dominant.*

Proof. We have $\mathrm{cycles}(M^{\oplus v}) \leq \mathrm{cycles}(M) + 1$. If S' is not dominant, meaning $\mathrm{cycles}(M^{\oplus S'}) < \mathrm{cycles}(M) + |S'|$, then $\mathrm{cycles}(M^{\oplus S})$ cannot "catch up" to $\mathrm{cycles}(M) + |S|$. □

Proof of Lemma 5.61. First, observe that any non-crossing subset of $\mathrm{gg}(M)$ is dominant. $v \in \mathrm{gg}(M)$ ensures that $M^{\oplus v}$ splits the cycle containing v into two cycles. Because the set of vertices is non-crossing, further splits will create one new cycle each time.

Now we show the converse. Let $S \subseteq V_4$ be dominant. There are three possibilities for $v \in V_4$:

(i) v is in two different cycles of M,

(ii) v occurs twice in the same cycle and is in $\mathrm{gg}(M)$,

(iii) v occurs twice in the same cycle and is not in $\mathrm{gg}(M)$.

In the first case, $\mathrm{cycles}(M^{\oplus v}) = \mathrm{cycles}(M) - 1$ decreases. Therefore $\{v\}$ is not dominant, and by Proposition 5.62, v cannot be in S. In the third case, $\mathrm{cycles}(M^{\oplus v}) = \mathrm{cycles}(M)$. Again, $\{v\}$ is not dominant and v cannot be in S. We deduce $S \subseteq \mathrm{gg}(M)$.

Next, we claim that a pair of crossing vertices $v, w \in \mathrm{gg}(C)$ are not dominant. We have $\mathrm{cycles}(M^{\oplus\{v,w\}}) = \mathrm{cycles}(M)$ whereas a dominant term should increase the cycle count by 2. Therefore, by Proposition 5.62 we conclude that S cannot contain any crossing pairs. This completes the proof of the lemma. □

Let s_M be the leading coefficient,

Definition 5.63. $s_M := \displaystyle\sum_{\textit{non-crossing } S \subseteq \mathrm{gg}(M)} (-1)^{|S|}.$

One step towards Conjecture 5.49 is to show that for planar graphs, an uncancelled term is always upper bounded by a simple matching. A concrete, purely combinatorial conjecture is the following,

Conjecture 5.64. *If G, H are graphs, $G \cup H$ is planar, $M \in \mathcal{PM}(G, H)$ and $s_M \neq 0$, then there is a simple matching $M' \in \mathcal{PM}(G, H)$ with* $\text{cycles}(M') \geq \text{cycles}(M)$.

It is easy to check that a simple $M \in \mathcal{PM}(G, H)$ is always uncancelled since $\text{cycles}(M^{\oplus S}) < \text{cycles}(M) + |S|$ holds with strict inequality for all $S \neq \emptyset$. If the above conjecture is true, terms with $s_M \neq 0$ will therefore be dominated by simple terms.

5.4 Polynomial Basis for the Boolean Setting

Let $H_n = \{-1, +1\}^n$. Letting $d_u \in_R H_n$, let Sym^V_{bool} be the set of polynomials p in the d_u which are symmetric under simultaneous automorphism of the hypercube: for any $\pi \in \text{Aut}(H_n)$,

$$p(d_1, \ldots, d_u, \ldots) = p(\pi d_1, \ldots, \pi d_u, \ldots).$$

$\text{Aut}(H_n)$ is well-known to be the hyperoctahedral group.

Fact 5.65. $\text{Aut}(H_n)$ *consists of permutations of the coordinates $[n]$ and bitflips using any $z \in \{-1, +1\}^n$. Formally,* $\text{Aut}(H_n)$ *is a semidirect product of S_n and $\mathbb{Z}_2^n$.*

We give a nice basis for such functions, showing formulas that mirror the general theme of routings and matchings in the underlying graph.

Definition 5.66 (Generalized inner product)**.** *For $d_1, \ldots, d_{2k}$, let*

$$\langle d_1, \ldots, d_{2k} \rangle = \sum_{i=1}^{n} d_{1,i} \cdots d_{2k,i}.$$

This is also denoted by the variable $x_{1,\ldots,2k}$.

We say that a hypergraph is even if the size of every hyperedge is even. Let $\deg_G(v)$ be the number of edges of G containing v. Given an even hypergraph G on vertex set $[m]$, let

$$m_G = \prod_{\{e_1,\ldots,e_{2k}\} \in E(G)} x_{e_1,\ldots,e_{2k}}.$$

Note that edges are allowed to repeat.

The m_G are not linearly independent. A basis is:

Lemma 5.67. *The set of m_G such that: there is $\sigma : E(G) \to [n]$ such that for all vertices $u \in V$ and edges $e, f \ni u$, $\sigma(e) \neq \sigma(f)$, is a basis for Sym^V_{bool}.*

Proof. Expand

$$m_G = \sum_{\sigma : E(G) \to [n]} \prod_{e = \{e_1, \ldots, e_{2k} \in E(G)\}} d_{e_1, \sigma(e)} \cdots d_{e_{2k}, \sigma(e)}.$$

If there is no such σ, then every term above has a square term $d_{ij}^2 = 1$. Therefore m_G simplifies to a lower-degree polynomial, and it can be expressed in terms of other m_G.

If there is a σ for G, then m_G contains a multilinear monomial with "shape" G, which is linearly independent from other m_G. More formally, to show linear independence, suppose $\sum_G c_G m_G = 0$ for some c_G not all zero. Taking a nonzero graph G with maximum number of edges, precisely the coefficient c_G appears on multilinear monomials with "shape" G, such as the monomial for σ, which is a contradiction. □

Corollary 5.68. *The set of m_G such that G has at most n hyperedges is linearly independent.*

Remark 5.69. *The hyperedges are sets, so they don't contain repeats (and thus G has no self-loops). If we did have an edge e with a repeated vertex i in G, we could delete two copies of i from e without affecting m_G because we always have that $d_{ij}^2 = 1$.*

As before, we can run Gram-Schmidt to orthogonalize the m_G. We will generalize matching collections to the Boolean case and use them to express the resulting polynomials p_G. In the Boolean case it is also useful to express p_G and various calculations as a sum over certain functions $\sigma : E(G) \to [n]$.

Definition 5.70. *Let $\mathcal{M}_{bool}(G)$ be the set of partitions of $E(G)$.*

Definition 5.71. *For $M \in \mathcal{M}_{bool}(G)$ define the routed hypergraph* route(M) *by replacing each block B by a single hyperedge containing $v \in V$ which are incident to an odd number of edges in B.*

Any block such that every $v \in V$ is incident to an even number of edges in B is called a "closed block". Closed blocks are deleted from route(M).

Definition 5.72. *For $M \in \mathcal{M}_{bool}(G)$ define the notation* cycles(M) *to be the number of closed blocks of the partition.*

Definition 5.73. *Let $\mathcal{PM}_{bool}(G)$ be the set of partitions of $E(G)$ such that every block is closed.*

Denote the falling and rising factorial by

$$x^{\underline{k}} := x(x-1)\cdots(x-k+1), \qquad x^{\overline{k}} := x(x+1)\cdots(x+k-1).$$

Lemma 5.74.

$$\mathbb{E}[m_G] = \sum_{\substack{\sigma : E(G) \to [n] \\ s.t.\ \forall i.\ \sigma^{-1}(i) \text{ even}}} 1 = \sum_{M \in \mathcal{PM}_{bool}(G)} n^{\underline{\text{cycles}(M)}}.$$

Proof. The first equality is obtained by expanding m_G into a sum of over all $\sigma : E(G) \to [n]$, then using linearity of expectation. The second equality is obtained by casing on which values of $\sigma(e)$ are equal, which induces a partition of $E(G)$. We have that σ contributes to the first sum if and only if all of the blocks of the induced partition are closed. Once the partition is fixed, there are $n^{\underline{\text{cycles}(M)}}$ ways to choose distinct values for each cycle. □

For now we give only one definition of p_G. The definition in terms of matchings is more complicated and is included in Section 5.4.1.

Definition 5.75 (Generic construction from Proposition 5.4)**.**

$$p_G = \sum_{\substack{\sigma:E(G)\to[n] \\ s.t.\ \forall e,f\ni u.\ \sigma(e)\neq\sigma(f)}} \prod_{e=\{e_1,\ldots,e_{2k}\}\in E(G)} d_{e_1,\sigma(e)} d_{e_2,\sigma(e)} \cdots d_{e_{2k},\sigma(e)}.$$

Lemma 5.76 (Automorphism-invariance)**.** $p_G \in Sym^V_{bool}$.

Proof. Neither of the two types of H_n symmetries changes p_G. Coordinate permutation doesn't change p_G because σ doesn't depend on the names of the coordinates. Bitflips don't change p_G because every hyperedge is even (so flips cancel out). □

Corollary 5.77. *p_G equals the output of the degree-orthogonal Gram-Schmidt process on the m_G.*

We can easily compute the inner product of p_G and p_H in the Boolean case. The idea is that p_G and p_H only contain terms where each vertex appears in each block at most once. When we multiply p_G and p_H together, these blocks may merge, giving us blocks where each vertex appears at most twice. If there is a block where a vertex appears only once, this block will have zero expected value, so the only terms which have nonzero expected value are the terms where in each block, each vertex either doesn't appear or appears twice, once from a G-edge and once from an H-edge. We now make this argument more precise.

Definition 5.78. *Let $\mathcal{PM}_{bool}(G,H)$ be the set of partitions of $E(G) \cup E(H)$ such that for each vertex and each block, the number of G-edges containing the vertex equals the number of H-edges.*

We say that a partition $M \in \mathcal{PM}_{bool}(G,H)$ is simple if for each block, each vertex appears at most 2 times.

Lemma 5.79.

$$\mathbb{E}[p_G p_H] = |\Sigma(G,H)| = \sum_{simple\ M\in\mathcal{PM}_{bool}(G,H)} n^{\underline{\text{cycles}(M)}}$$

where $\Sigma(G,H)$ is the set of functions $\sigma : E(G \cup H) \to [n]$ such that

(i) For $e,f \in E(G)$ such that $e \cap f \neq \emptyset$, $\sigma(e) \neq \sigma(f)$.

(ii) For $e,f \in E(H)$ such that $e \cap f \neq \emptyset$, $\sigma(e) \neq \sigma(f)$.

(iii) For all u,i, the size of $\{u \in e \in E(G \cup H) : \sigma(e) = i\}$ is even. Note that from conditions (i) and (ii) it must be size either 0 or 2.

Proof. The first equality follows from expanding p_G, p_H and using linearity of expectation. The second equality follows from looking at the partition induced by σ. The definition of $\Sigma(G,H)$ exactly checks that this partition is simple and in $\mathcal{PM}_{bool}(G,H)$. □

Corollary 5.80. $n^{\underline{|E(G)|}} \leq \mathbb{E}[p_G^2] \leq (2|E(G)|)^{2|E(G)|} n^{\underline{|E(G)|}}$.

Proof. Each cycle in M requires at least two edges, and hence the maximum magnitude is bounded by $n^{\underline{|E(G)|}}$. Furthermore, this can be achieved by matching each edge with its duplicate. The number of partitions of a k-element set is at most k^k, which proves the upper bound. □

The inner product formula implies that all inner products are non-negative, so the Boolean case does not exhibit the "negative inner product" abnormality of the spherical case with the K_5 example.

5.4.1 Formulas using the partition poset

Since p_G is automorphism-invariant, it can be expressed in terms of the m_G basis. However, the coefficients on the m_G are not that easy to work with.

Definition 5.81 (Routing definition)**.**

$$p_G = \sum_{M \in \Lambda_G^c} \mu(\varnothing, M) n^{\text{cycles}(M)} m_{\text{route}(M)}$$

where μ is the Möbius function of the poset Λ_G^c (to be defined in Definition 5.86).

These coefficients can be computed by an inclusion-exclusion recurrence (which is in truth computing the Möbius function of a poset based on G, see [Sta12, Chapter 3] for an overview of poset combinatorics).

Example 5.82. *Let G have four parallel edges $\{s, t\}$. Using Definition 5.75,*

$$p_G = \sum_{\textit{injective } \sigma:[4]\to[n]} d_{s,\sigma(1)} d_{t,\sigma(1)} d_{s,\sigma(2)} d_{t,\sigma(2)} d_{s,\sigma(3)} d_{t,\sigma(3)} d_{s,\sigma(4)} d_{t,\sigma(4)}.$$

The leading monomial is $\langle d_s, d_t \rangle^4$. Subtract off terms where two edges are given the same label,

$$\binom{4}{2} n \langle d_s, d_t \rangle .$$

This puts a coefficient of -2 on terms with three equal labels and one unequal label. Add them back,

$$\binom{4}{1} 2 \langle d_s, d_t \rangle^2 .$$

The coefficient of terms with two pairs of two equal labels is -1. Add them back,

$$3 \cdot n^2.$$

Finally, the coefficient of the all-equal label is now 6. Subtract out

$$6n.$$

In total,

$$p_G = \langle d_s, d_t \rangle^4 - \binom{4}{2} n \langle d_s, d_t \rangle^2 + \binom{4}{1} 2 \langle d_s, d_t \rangle^2 + 3n^2 - 6n.$$

Definition 5.83. *Let Λ_G be the partition poset of $E(G)$: the elements are partitions of $E(G)$, and $M_1 \preceq M_2$ if M_1 refines M_2.*

Λ_G has a unique minimal element (the partition into singletons, to be denoted by $\emptyset$) and a unique maximal element (the partition with one block). The example above corresponds to the standard partition poset of $\{1, 2, 3, 4\}$ [Sta12, Example 3.10.4].

Observe that the poset refinement relation exactly captures how several coefficients on m_G can contribute to the same coefficient on d. Stated formally, for $\sigma : E(G) \to [n]$ let $M(\sigma)$ denote the partition of $E(G)$ induced by σ. Then

Fact 5.84. *Given $\lambda : \Lambda_G \to \mathbb{R}$,*

$$\sum_{M \in \Lambda_G} \lambda(M) n^{\text{cycles}(M)} m_{\text{route}(M)} = \sum_{\sigma : E(G) \to [n]} \left(\sum_{M' \preceq M(\sigma)} \lambda(M') \right) \prod_{e = \{e_1, \ldots, e_{2k}\} \in E(G)} d_{e_1, \sigma(e)} \cdots d_{e_{2k}, \sigma(e)}.$$

Inverting the coefficients can be done by Möbius inversion.

Lemma 5.85. *Let $\kappa \subseteq \Lambda_G$ be downward-closed. Let $\overline{\kappa} = \{\emptyset\} \cup (\Lambda_G \setminus \kappa)$. Then*

$$\sum_{\substack{\sigma : E(G) \to [n] \\ \text{s.t. } M(\sigma) \in \kappa}} \prod_{e = \{e_1, \ldots, e_{2k}\} \in E(G)} d_{e_1, \sigma(e)} \cdots d_{e_{2k}, \sigma(e)} = \sum_{M \in \overline{\kappa}} \mu(\emptyset, M) n^{\text{cycles}(M)} m_{\text{route}(M)}$$

where μ is the Möbius function of $\overline{\kappa}$.

Definition 5.86. *Let Λ_G^c be the set of partitions of $E(G)$ such that either the partition is $\emptyset$ or there is a block and a vertex such that there are at least 2 edges of the block containing the vertex.*

$\Lambda_G^c = \overline{\kappa}$ where κ is the (downward-closed) defining set of partitions for p_G in Definition 5.75. In summary from Lemma 5.85 we have,

Lemma 5.87. *Definition 5.81 is equivalent to Definition 5.75.*

There is also a "Boolean Isserlis theorem". The Boolean Isserlis theorem allows us to compute for fixed $d_{ij} \in \mathbb{R}^n$ and $v \in_{\text{R}} \{-1, +1\}^n$,

$$\mathop{\mathbb{E}}_{v \in_{\text{R}} \{-1,+1\}^n} [\langle v, d_{1,1}, \ldots, d_{1,\ell_1} \rangle \langle v, d_{2,1}, \ldots, d_{2,\ell_2} \rangle \cdots \langle v, d_{k,1}, \ldots, d_{k,\ell_k} \rangle].$$

Combining together the d_{ij} component-wise, it suffices to compute

$$\mathop{\mathbb{E}}_{v \in_R \{-1,+1\}^n} [\langle v, d_1 \rangle \langle v, d_2 \rangle \cdots \langle v, d_k \rangle].$$

Let Λ_{2k} be the partition poset for $2k$ elements and Λ_{2k}^e the subset where each block has even size.

Lemma 5.88 (Boolean Isserlis theorem). *For fixed $d_i \in \mathbb{R}^n$ and $v \in_R \{-1,+1\}^n$,*

$$\mathop{\mathbb{E}}_{v \in_R \{-1,+1\}^n} [\langle v, d_1 \rangle \langle v, d_2 \rangle \cdots \langle v, d_{2k} \rangle] = \sum_{\substack{\sigma : [2k] \to [n] \\ \text{s.t. } \forall i.\ |\sigma^{-1}(i)| \text{ even}}} d_{1,\sigma(1)} \cdots d_{2k,\sigma(2k)}$$
$$= \sum_{M \in \Lambda_{2k}^e} \lambda(M) \prod_{\{e_1, \ldots, e_\ell\} \in M} \langle d_{e_1}, \ldots, d_{e_\ell} \rangle$$

where $\lambda : \Lambda_{2k}^e \to \mathbb{R}$ is defined by the recursion

$$\lambda(\textit{perfect matching}) = 1,$$
$$\sum_{M' \preceq M} \lambda(M') = 1.$$

Proof. The first equality is by expanding and applying linearity of expectation. The second equality sums $\lambda(M)$ in a way such that each coefficient on the relevant d is 1 or 0. The function $\lambda : \Lambda_{2k} \to \mathbb{R}$ needs to satisfy

$$\forall M \in \Lambda_{2k}^e.\ \sum_{M' \preceq M} \lambda(M') = 1, \qquad \forall M \in \Lambda_{2k} \setminus \Lambda_{2k}^e.\ \sum_{M' \preceq M} \lambda(M') = 0.$$

There is a unique function, given by Möbius inversion on Λ_{2k},

$$\lambda(M) = \begin{cases} \sum_{\substack{M' \preceq M: \\ M' \in \Lambda_{2k}^e}} \mu(M', M) & M \in \Lambda_{2k}^e \\ 0 & M \notin \Lambda_{2k}^e \end{cases}$$

where $\mu(M', M)$ is the Möbius function for Λ_{2k} given in [Sta12, Example 3.10.4]. Equivalently, $\lambda(M)$ must equal the recursion given in the lemma statement. □

Examples:

$$\begin{aligned}
\mathbb{E}[\langle v, d_1\rangle \langle v, d_2\rangle] &= \langle d_1, d_2\rangle \\
\mathbb{E}[\langle v, d_1\rangle \langle v, d_2\rangle \langle v, d_3\rangle \langle v, d_4\rangle] &= \langle d_1, d_2\rangle \langle d_3, d_4\rangle + \langle d_1, d_3\rangle \langle d_2, d_4\rangle + \langle d_1, d_4\rangle \langle d_2, d_3\rangle \\
&\quad - 2\langle d_1, d_2, d_3, d_4\rangle \\
\mathbb{E}[\prod_{i=1}^{6} \langle v, d_i\rangle] &= \langle d_1, d_2\rangle \langle d_3, d_4\rangle \langle d_5, d_6\rangle + 15 \text{ terms of type (2,2,2)} \\
&\quad - 2 \langle d_1, d_2\rangle \langle d_3, d_4, d_5, d_6\rangle + \binom{6}{2} \text{ terms of type (2,4)} \\
&\quad + 16\langle d_1, d_2, d_3, d_4, d_5, d_6\rangle \\
\mathbb{E}[\prod_{i=1}^{8} \langle v, d_i\rangle] &= \langle d_1, d_2\rangle \langle d_3, d_4\rangle \langle d_5, d_6\rangle \langle d_7, d_8\rangle + 105 \text{ terms of type (2,2,2,2)} \\
&\quad - 2 \langle d_1, d_2\rangle \langle d_3, d_4\rangle \langle d_5, d_6, d_7, d_8\rangle + 3\binom{8}{4} \text{ terms of type (2,2,4)} \\
&\quad + 16 \langle d_1, d_2\rangle \langle d_3, d_4, d_5, d_6, d_7, d_8\rangle + \binom{8}{2} \text{ terms of type (2,6)} \\
&\quad + 4\langle d_1, d_2, d_3, d_4\rangle\langle d_5, d_6, d_7, d_8\rangle + \frac{1}{2}\binom{8}{4} \text{ terms of type (4,4)} \\
&\quad + 8\langle d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8\rangle
\end{aligned}$$

5.5 Inversion Formula for Approximate Orthogonality

Consider the problem of Fourier inversion: given parameters $\widehat{f}(G)$ for different graphs G, find an orthogonally invariant function $f : (\mathbb{R}^n)^V \to \mathbb{R}$ such that

$$\langle f, p_G\rangle = \widehat{f}(G).$$

If the p_G were completely orthogonal, then the function

$$f = \sum_G \widehat{f}(G) \cdot \frac{p_G}{\mathbb{E}\, p_G^2}$$

is the unique f in the span of p_G for given G. In general, let Q be the square matrix indexed by graphs G with entries $Q[G, H] := \langle p_G, p_H\rangle$. Then f is given by

$$f = \sum_G \left(\sum_H Q^{-1}[G, H] \cdot \widehat{f}(H) \right) \cdot p_G$$

provided that Q is invertible.

Because of approximate orthogonality, Q is close to a diagonal matrix. Therefore Q^{-1}

is also close to a diagonal matrix. Formally we show

Lemma 5.89. *Suppose we are in either the Gaussian, spherical, or Boolean setting. Let finitely many nonzero $\widehat{f}(G) \in \mathbb{R}$ be given where G is a graph of the appropriate type for the setting, and assume that $|E(G)| = o\left(\frac{\log n}{\log \log n}\right)$ for all given G. For sufficiently large n, there is a unique f satisfying $\langle f, p_G \rangle = \widehat{f}(G)$, and f equals*

$$f = \sum_H \left(\widehat{f}(H) + o(1) \cdot \max_{G \leftrightarrow H} \left| \widehat{f}(G) \right| \right) \cdot \frac{p_H}{\mathbb{E}\, p_H^2}.$$

Proof. Since the p_G are orthogonal if $G \not\leftrightarrow H$, the matrix Q is block diagonal with blocks defined by $\leftrightarrow$. The bound on the size of given G implies that the dimension of each block is $n^{o(1)}$.

The diagonal terms are

$$\mathbb{E}[p_G^2] = \begin{cases} n^{|E(G)|+o(1)} & \text{Gaussian case (Corollary 5.28)} \\ n^{-|E(G)|+o(1)} & \text{Spherical case (Corollary 5.46)} \\ n^{|E(G)|+o(1)} & \text{Boolean case (Corollary 5.80)} \end{cases}.$$

The off-diagonal terms with $G \leftrightarrow H$ are bounded by

$$|\mathbb{E}[p_G p_H]| \le \begin{cases} \max\limits_{M \in \mathcal{PM}(G,H)} n^{\text{cycles}(M)+o(1)} & \text{Gaussian case (Lemma 5.26)} \\ n^{-2|E(G)|} \max\limits_{M \in \mathcal{PM}(G,H)} n^{\text{cycles}(M)+o(1)} & \text{Spherical case (Corollary 5.47)} \\ \max\limits_{\text{simple } M \in \mathcal{PM}_{bool}(G,H)} n^{\text{cycles}(M)+o(1)} & \text{Boolean case (Lemma 5.79)} \end{cases}.$$

When $G \neq H$, we claim that cycles(M) must be strictly less than $|E(G)|$. Any $M \in \mathcal{PM}(G, H)$ achieving $|E(G)|$ cycles must pair up edges of G and H, which shows the contrapositive.

Therefore the off-diagonal terms are smaller by a factor of $n^{1-o(1)}$ than the diagonal term. Therefore Q is invertible (for sufficiently large n) and

$$Q^{-1}[G, H] = \begin{cases} \frac{1}{\mathbb{E}[p_G^2]} & G = H \\ n^{o(1)-1} \cdot \frac{1}{\mathbb{E}[p_G^2]} & G \neq H \end{cases}.$$

This proves the approximate Fourier inversion. □

Remark 5.90. *Using more careful counting, the assumption on $|E(G)|$ can likely be improved to $|E(G)| \le n^\delta$ for some explicit $\delta > 0$.*

5.6 Open Problems

It remains to better understand the inner product polynomials in the spherical setting. The connection to graph planarity is interesting but remains conjectural, Conjecture 5.49. It's possible that a different formula for p_G in the spherical setting could clarify the inner product calculation $\mathbb{E}[p_G \cdot p_H]$.

$\mathbb{E}[p_G \cdot p_H]$ in the Boolean setting (Lemma 5.79) and spherical setting (Conjecture 5.49) is determined by matchings between G and H such that the cycles are *simple*. Computing the maximum number of cycles among such matchings is NP-hard, Remark 5.27. We conjecture that checking whether or not a matching with simple cycles exists is already NP-hard.

Conjecture 5.91. *The following problem is NP-hard: given an undirected graph G where each edge is either red or blue, can you partition $E(G)$ into simple cycles that alternate color?*

Conjecture 5.92. *The following problem is NP-hard: given an undirected graph G, can you partition $E(G)$ into simple cycles with even lengths?*

An interesting recent algorithm was given by Björklund et al to find the *shortest* even-length cycle in a directed graph [BHK21].

CHAPTER 6
LOWER BOUND FOR SPARSE INDEPENDENT SET

We initiate the study of the sum-of-squares algorithm on random, *sparse* inputs. As a case study, we analyze the performance of sum-of-squares on the independent set problem.

For the dense case, average-case independent set is equivalent to finding a clique and the paper [BHK$^+$16] shows an average-case lower bound against the sum-of-squares algorithm. We extend the techniques introduced there, namely pseudocalibration, graph matrices, and the approximate decomposition into positive semidefinite matrices, in order to show that higher-degree sum-of-squares does not significantly beat the simple spectral relaxation, the Lovász ϑ-function.

In previous sum-of-squares lower bounds, such as those in Chapter 4, the input is a collection of iid random variables whose distribution is independent of n. For example, on Constraint Satisfaction Problems, one fixes the structure of the instance and only considers an instance to be specified by the signs of the literals, which can be viewed as uniformly random $\{-1, +1\}$ variables. In the sparse setting, the distribution of each variable may depend on n, such as in $G_{n,p}$ with $p = o(1)$.

At its core, our proof uses the "pseudocalibration plus graph matrices" analysis introduced in Chapter 3, but we need several significant extensions in the sparse setting. For one, the pseudocalibration heuristic requires a planted distribution that is indistinguishable from the input distribution via low-degree tests. In the sparse case the natural planted distribution *does* admit low-degree distinguishers (counting small subgraphs), and we show how to adapt pseudocalibration to overcome this.

Graph matrix analysis in the sparse setting also requires a significant update. As overviewed in Section 2.2, norm bounds for graph matrices on $G_{n,p}$ depend on a different type of minimum vertex separator than in the dense case. With these norm bounds in hand, we perform an approximate PSD decomposition as described in Section 3.3, though there are several additional complications.

The layout of this chapter is as follows. In Section 6.1, we state the results. The additional proof techniques for the sparse setting are outlined at a high level in Section 6.2. The "connected truncation" fix for pseudocalibration is described in Section 6.3. The PSDness proof is executed in Section 6.4.

Bibliography. This chapter is the main content of the paper [JPR$^+$21]. The high-level technical tools were extracted out into Chapter 2 and Chapter 3. The proof outline in Section 6.2 is organized in a slightly different way than in the paper. A new overview of the combinatorial arguments is provided in Section 6.4.2. Some technical components of the proof have been omitted.

6.1 Statement of results

Sample $G \sim G_{n,\frac{d}{n}}$ as an Erdős-Rényi random graph[1] with average degree d, where we think of $d \ll n$. Specializing to the problem of independent set, a maximum independent set in G has size:

Fact 6.1 ([COE15, DM11, DSS16]). *W.h.p. the max independent set in G has size* $(1 + o_d(1)) \cdot \frac{2 \ln d}{d} \cdot n$.

The value of the degree-2 SoS relaxation for independent set equals the Lovász ϑ function, which is an upper bound on the independence number $\alpha(G)$, by virtue of being a relaxation. For random graphs $G \sim G_{n,d/n}$ this value is larger by a factor of about $\sqrt{d}$ than the true value of $\alpha(G)$ with high probability.

Fact 6.2 ([CO05]). *W.h.p.* $\vartheta(G) = \Theta(\frac{n}{\sqrt{d}})$.

We will prove that the value of higher-degree SoS is also on the order of $n/\sqrt{d}$, rather than n/d, and thereby demonstrate that the information-computation gap against basic SDP/spectral algorithms persists against higher-degree SoS.

In considering relaxations for independent set of a graph $G = (V, E)$, with variables x_v being the 0/1 indicators of the independent set, the SoS relaxation searches for pseudoexpectation operators satisfying the polynomial constraints

$$\forall v \in V. \quad x_v^2 = x_v \qquad \text{and} \qquad \forall (u,v) \in E. \quad x_u x_v = 0\,.$$

The objective value of the convex relaxation is given by the quantity $\widetilde{\mathbb{E}}[\sum_{v \in V} x_v] = \sum_{v \in V} \widetilde{\mathbb{E}}[x_v]$. The following theorem states our main result.

Theorem 6.3. *There is an absolute constant $c_0 \in \mathbb{N}$ such that for sufficiently large $n \in \mathbb{N}$ and $d \in [(\log n)^2, n^{0.5}]$, and parameters k, D_{SoS} satisfying*

$$k \le \frac{n}{D_{\text{SoS}}^{c_0} \cdot \log n \cdot d^{1/2}}\,,$$

it holds w.h.p. for $G = (V, E) \sim G_{n,\, d/n}$ that there exists a degree-D_{SoS} pseudoexpectation satisfying

$$\forall v \in V. \quad x_v^2 = x_v \qquad \textit{and} \qquad \forall (u,v) \in E. \quad x_u x_v = 0\,,$$

and objective value $\widetilde{\mathbb{E}}[\sum_{v \in V} x_v] \ge (1 - o(1))k$.

Remark 6.4. *This is a non-trivial lower bound whenever $D_{\text{SoS}} \le \left(\frac{d^{1/2}}{\log n}\right)^{1/c_0}$.*

Remark 6.5. *It suffices to set $c_0 = 20$ for our current proof. We did not optimize the tradeoff in D_{SoS} with k, but we did optimize the log factor (with the hope of eventually removing it).*

[1] Unfortunately our techniques do not work for a random d-regular graph. See Section 6.5.

Remark 6.6. *Using the same technique, we can prove an $n^{\Omega(\varepsilon)}$ SoS-degree lower bound for all $d \in [\sqrt{n}, n^{1-\varepsilon}]$.*

For $n^{\varepsilon} \le d \le n^{0.5}$, the theorem gives a polynomial n^{δ} SoS-degree lower bound. For smaller d, the bound is still strong against low-degree SoS, but it becomes trivial as D_{SoS} approaches $(d^{1/2}/\log n)^{1/c_0}$ or d approaches $(\log n)^2$ since k matches the size of the maximum independent set in G, hence there is an actual distribution over independent sets of this size (the expectation operator for which is trivially is also a pseudoexpectation operator).

The above bound says nothing about the "almost dense" regime $d \in [n^{1-\varepsilon}, n/2]$. To handle this regime, we observe that our techniques, along with the ideas from the $\Omega(\log n)$-degree SoS bound from [BHK+16] for the dense case, prove a lower bound for any degree $d \ge n^{\varepsilon}$.

Theorem 6.7. *For any $\varepsilon_1, \varepsilon_2 > 0$ there is $\delta > 0$, such that for $d \in [n^{\varepsilon_1}, n/2]$ and $k \le \frac{n}{d^{1/2+\varepsilon_2}}$, it holds w.h.p. for $G = (V, E) \sim G_{n,\, d/n}$ that there exists a degree-$(\delta \log d)$ pseudoexpectation satisfying*

$$\forall v \in V. \quad x_v^2 = x_v \qquad \text{and} \qquad \forall (u,v) \in E. \quad x_u x_v = 0\,,$$

and objective value $\widetilde{\mathbb{E}}[\sum_{v \in V} x_v] \ \ge\ (1 - o(1))k$.

In particular, these theorems rule out polynomial-time certification (i.e. constant degree SoS) for any $d \ge \text{polylog}(n)$.

6.1.1 Related work

The average-case independent set problem is a sparse version of the Planted Clique problem, and our proof technique extends the SoS lower bound for Planted Clique by Barak et al [BHK+16].

For the case of independent set in random sparse graphs, many works have considered the *search* problem of finding a large independent set in a random sparse graph. Graphs from $G_{n,d/n}$ are known to have independent sets of size $(2 + o_d(1)) \cdot \frac{\ln d}{d} \cdot n$ with high probability, and it is possible to find an independent set of size $(1 + o_d(1)) \cdot \frac{\ln d}{d} \cdot n$, either by greedily taking a maximal independent set in the dense case [GM75] or by using a local algorithm in the sparse case [Wor95]. This is conjectured to be a computational phase transition, with concrete lower bounds against search beyond $\frac{\ln d}{d} \cdot n$ for local algorithms [RV17a] and low-degree polynomials [Wei20]. The game in the search problem is all about the constant 1 vs 2, whereas our work shows that the integrality gap of SoS is significantly worse, on the order of $\sqrt{d}$. Lower bounds against search work in the regime of constant d (though in principle they could be extended to at least some $d = \omega(1)$ with additional technical work), while our techniques require $d \ge \log(n)$. For search problems, the *overlap distribution* of two high-value solutions has emerged as a heuristic indicator of computational hardness, whereas for certification problems it is unclear how the overlap distribution plays a role.

Norm bounds for sparse graph matrices were also obtained using a different method of matrix deviation inequalities by Rajendran and Tulsiani [RT20].

The work [BBK$^+$21b] constructs a *computationally quiet* planted distribution that is a candidate for pseudocalibration. However, their distribution is not quite suitable for our purposes. [2] [3]

A recent paper by Pang [Pan21] fixes a technical shortcoming of [BHK$^+$16] by constructing a pseudoexpectation operator that satisfies "$\sum_{v \in V} x_v = k$" as a polynomial constraint (whereas the shortcoming was $\widetilde{\mathbb{E}}[\sum_{v \in V} x_v] \geq (1 - o(1))k$ like we have here).

6.2 Proof Outline

Since SoS is a convex program, the goal of an SoS lower bound is to construct a dual object, which is a feasible pseudoexpectation operator. We will follow the general approach to SoS lower bounds outlined in Chapter 3. We use the p-biased Fourier basis (Section 6.2.1), pseudocalibrate with an additional "connected truncation" (Section 6.2.2), then perform graph matrix analysis as described in Section 6.2.3. The key charging arguments for the approximate PSD decomposition are given in Section 6.2.4.

The sparse setting requires many new ideas which are outlined in this section, leaving the technical details to later sections. The ideas here without the additional technical details are almost enough to formally prove Theorem 6.7, as we describe in Section 6.2.5.

6.2.1 *p-biased Fourier analysis and graph matrices*

Since we are interested in sparse Erdős-Rényi graphs in this work, we will resort to p-biased Fourier analysis [O'D14, Section 8.4]. Formally, we view the input graph $G \sim G_{n,p}$ as a vector in $\{0,1\}^{\binom{n}{2}}$ indexed by sets $\{i,j\}$ for $i,j \in [n], i \neq j$, where each entry is independently sampled from the p-biased Bernoulli distribution, Bernoulli(p). Here, by convention $G_e = 1$ indicates the edge e is present, which happens with probability p. The Fourier basis we use for analysis on G is the set of p-biased Fourier characters (which are naturally indexed by graphs H on $[n]$).

Definition 6.8. *χ denotes the p-biased Fourier character,*

$$\chi(0) = \sqrt{\frac{p}{1-p}}, \qquad \chi(1) = -\sqrt{\frac{1-p}{p}}.$$

[2] [BBK$^+$21b] provide evidence that their distribution is hard to distinguish from $G_{n,d/n}$ with probability $1 - o(1)$ (it is not "strongly detectable"). However, their distribution *is* distinguishable with probability $\Omega(1)$, via a triangle count (it is "weakly detectable"). In SoS pseudocalibration, this manifests as $\widetilde{\mathbb{E}}[1] = \Theta_d(1)$. We would like the low-degree distinguishing probability to be $o(1)$ i.e. $\widetilde{\mathbb{E}}[1] = 1 + o_d(1)$ so that normalizing by $\widetilde{\mathbb{E}}[1]$ does not affect the objective value.

[3] Another issue is that their planted distribution introduces noise by adding a small number of edges inside the planted independent set.

For H a subset or multi-subset of $\binom{[n]}{2}$, let $\chi_H(G) := \prod_{e\in H}\chi(G_e)$.

We will also need the function $1 - G_e$ which indicates that an edge is not present.

Definition 6.9. *For $H \subseteq \binom{[n]}{2}$, let $1_{\overline{H}}(G) = \prod_{e\in H}(1 - G_e)$.*

When H is a clique, this is the *independent set indicator* for the vertices in H.

Proposition 6.10. *For $e \in \{0,1\}$, $1 + \sqrt{\frac{p}{1-p}}\chi(e) = \frac{1}{1-p}(1-e)$. Therefore, for any $H \subseteq \binom{[n]}{2}$,*

$$\sum_{T\subseteq H}\left(\frac{p}{1-p}\right)^{|T|/2}\chi_T(G) = \frac{1}{(1-p)^{|H|}}\cdot 1_{\overline{H}}(G).$$

We now define the graph matrices that we use. The definitions will need to be modified slightly during the PSD-ness proof. See Section 6.4.

Definition 6.11 (Ribbon)**.** *A ribbon is a tuple $R = (V(R), E(R), A_R, B_R)$, where $(V(R), E(R))$ is an undirected multigraph without self-loops, $V(R) \subseteq [n]$, and $A_R, B_R \subseteq V(R)$. Let $C_R := V(R) \setminus (A_R \cup B_R)$.*

Definition 6.12 (Matrix for a ribbon)**.** *For a ribbon R, the matrix M_R has rows and columns indexed by all subsets of $[n]$ and has a single nonzero entry,*

$$M_R[I,J] = \begin{cases}\chi_{E(R)}(G) & I = A_R, J = B_R\\ 0 & \textit{Otherwise}\end{cases}$$

Definition 6.13 (Ribbon isomorphism)**.** *Two ribbons R, S are isomorphic, or have the same shape, if there is a bijection between $V(R)$ and $V(S)$ which is a multigraph isomorphism between $E(R), E(S)$ and is a bijection from A_R to A_S and B_R to B_S. Equivalently, letting S_n permute the vertex labels of a ribbon, the two ribbons are in the same S_n-orbit.*

Definition 6.14 (Shape)**.** *A shape is an equivalence class of ribbons with the same shape. Each shape has associated with it a representative $\alpha = (V(\alpha), E(\alpha), U_\alpha, V_\alpha)$, where $U_\alpha, V_\alpha \subseteq V(\alpha)$. Let $W_\alpha := V(\alpha) \setminus (U_\alpha \cup V_\alpha)$.*

Definition 6.15 (Graph matrix)**.** *For a shape α, the graph matrix M_α is*

$$M_\alpha = \sum_{\textit{injective } \phi: V(\alpha)\to[n]} M_{\phi(\alpha)}.$$

6.2.2 Modifying pseudocalibration

Failure of pseudocalibration The first obstacle we overcome is the lack of a planted distribution. Pseudocalibration requires a planted and random distribution which are hard to distinguish using the *low-degree, likelihood ratio test* (i.e. $\widetilde{\mathbb{E}}[1]$ is bounded whp) [Hop18]. In the case of sparse independent set, we have the following natural hypothesis testing problem with which one may hope to pseudocalibrate.

- Null Hypothesis: Sample a graph $G \sim G_{n,p}$.
- Alternate Hypothesis: Sample a graph $G \sim G_{n,p}$. Then, sample a subset $S \subseteq [n]$ where each vertex is chosen with probability $\frac{k}{n}$. Then, plant an independent set in S, i.e. remove all the edges inside S.

In the case of sparse independent set, the naïve planted distribution *is* distinguishable from a random instance via a simple low-degree test – counting 4-cycles. In all uses of pseudocalibration that we are aware of, the two distributions being compared are conjecturally hard to distinguish by all polynomial-time algorithms. We are still searching for a suitable planted distribution for sparse independent set, and we believe this is an interesting question on its own.

Fixing pseudocalibration via connected truncation To get around with this issue, we close our eyes and "pretend" the planted distribution is quiet, ignoring the obvious distinguisher, and make a "connected truncation" of the moment matrix to remove terms which correspond to counting subgraphs in G. What remains is that $\widetilde{\mathbb{E}}[x^S]$ is essentially independent of the global statistics of G. It should be pointed out here that this is inherently distinct from the local truncation for weaker hierarchies (e.g. Sherali-Adams) where the moment matrix is an entirely local function [CM18]. In contrast, our $\widetilde{\mathbb{E}}[x^S]$ may depend on parts of the graph that are far away from S, in fact, even up to radius n^{δ}, exceeding the diameter of the random graph!

More formally, the candidate moment matrix Λ will be written as follows.

$$\begin{aligned}\lambda_\alpha &:= \left(\frac{k}{n}\right)^{|V(\alpha)| - \frac{|U_\alpha| + |V_\alpha|}{2}} \cdot \left(\frac{p}{1-p}\right)^{\frac{|E(\alpha)|}{2}} \\ \Lambda &:= \sum_{\alpha \in \mathcal{S}} \lambda_\alpha \cdot M_\alpha \,.\end{aligned}$$

We still use the natural planted distribution to compute the pseudocalibrated Fourier coefficients λ_α. However, we also truncate (set to zero) several Fourier coefficients. $\mathcal{S}$ ranges over all proper shapes α of appropriately bounded size such that *all vertices of α are connected to $U_\alpha \cup V_\alpha$*. The latter property is the important distinction from standard pseudocalibration and is what we call "connected truncation". This is perhaps the most conceptually interesting part of the proof, and we hope that the same "connected truncation" will be useful for other integrality gap constructions. Further discussion of the connected truncation is in Section 6.3.

6.2.3 Graph matrix analysis

Once the candidate moment matrix is written as a sum of graph matrices, we can analyze the sum to prove that it is PSD with high probability.

Norm bounds To bound error terms in the analysis, we need to understand the spectral norm of a sparse graph matrix M_α with high probability.

Existing norm bounds for graph matrices from [AMP20] may be applied to $G_{n,p}$, but unfortunately, they're too weak to be useful. Consider the case where we sample $G \sim G_{n,p}$ and try to bound the spectral norm of the one-edge shape, i.e. the centered and normalized adjacency matrix $A \in \mathbb{R}^{n \times n}$ whose entries are:

$$A[i,j] = \begin{cases} -\sqrt{\frac{1-p}{p}} & \{i,j\} \in E(G) \\ \sqrt{\frac{p}{1-p}} & \{i,j\} \notin E(G) \end{cases}.$$

Existing norm bounds give a bound of $\widetilde{\mathrm{O}}(\frac{\sqrt{n(1-p)}}{\sqrt{p}})$ whereas the true norm is $O(\sqrt{n})$ regardless of d or p. These differ by a $\mathrm{poly}(n)$ factor when p is sufficiently small, whereas we would like to describe the norm up to subpolynomial factors (or better). This discrepancy is even more pronounced when we use shapes with more vertices.

Using the trace method, in the full version of the paper we prove the norm bound:

$$\|M_\alpha\| \leq \widetilde{\mathrm{O}}\left(\max_{\substack{\text{vertex separators} \\ S \text{ of } \alpha}} \sqrt{n}^{|V(\alpha)|-|S|} \sqrt{\frac{1-p}{p}}^{|E(S)|} \right).$$

The formal statement is given in Section 6.4.9. As described in Remark 2.19, this norm bound is not always tight and could be improved.

When $p = \frac{1}{2}$, this bound recovers the norm bounds for $G_{n,1/2}$ up to lower order factors, as in this case the minimizing separator is the smallest one. The key conceptual takeaway is that, in the sparse setting, we need to redefine the weight of a vertex separator to also incorporate the edges within the separator, as opposed to only considering the number of vertices. We clearly distinguish these with the terms *Dense Minimum Vertex Separator* (DMVS) and *Sparse Minimum Vertex Separator* (SMVS).

Approximate PSD decomposition We then perform an approximate PSD decomposition of the graph matrices that make up Λ. The general factoring strategy is the same as in Section 3.3, though in the sparse regime we must be very careful about what kind of combinatorial factors we allow. Each shape comes with a natural "vertex decay" coefficient arising from the fractional size of the independent set and an "edge decay" coefficient arising from the sparsity of the graph. The vertex decay coefficients can be analyzed in a method similar to Planted Clique (which only has vertex decay). For the edge decay factors, we use novel charging arguments given in the next subsection.

The moment matrix actually must have a null space, due to the independent set constraints. We are forced to include certain shapes that encode indicator functions of independent sets. These shapes must be factored out and tracked separately throughout the analysis, in a similar way to the clique constraints in Planted Clique [BHK+16].

At this point, the techniques are strong enough to prove Theorem 6.7, an SoS-degree $\Omega(\log n)$ lower bound for $d \geq n^{\varepsilon}$, as we explain in Section 6.2.5. The remaining techniques are needed to push the SoS degree up and the graph degree down.

Throwing away dense shapes In our analysis, it turns out that a norm bound from the vanilla trace method is not quite sufficient. Sparse random matrices' spectral norms are fragile with respect to the influence of an unlikely event, exhibiting deviations away from the expectation with polynomially small probability (rather than exponentially small probability, like what is obtained from a good concentration bound). These "bad events" are small dense subgraphs present in a graph sampled from $G_{n,p}$.

In practice, this means that our trace method-based norm bound is not tight enough for shapes with lots of edges (these are the shapes that will "detect" the rare, dense subgraphs). A second and more subtle problem with these dense shapes[4] is that there are too many of them; combinatorial factors will overwhelm the "charging" argument. We utilize two methods to get rid of these dense shapes, so that we can restrict our sums to only sparse shapes.

Method #1: conditioning The first way to handle dense shapes is to condition on the high probability event that G has no small dense subgraphs. For example, for $d = n^{1-\varepsilon}$ whp every small subgraph S has $O(|S|)$ edges (even up to size n^{δ}). For a **shape** which is dense (i.e. v vertices and more than $O(v)$ edges) we can show that its norm falls off extremely rapidly under this conditioning, allowing us to throw away dense shapes.

This type of conditioning is well-known: a long line of work showing tight norm bounds for the simple adjacency matrix appeals to a similar conditioning argument within the trace method [BLM15, Bor15, FM17, DMO+19]. The conditioning is instantiated through the following identity.

Observation 6.16. *Given a set of edges $E \subseteq \binom{[n]}{2}$, if we know that not all of the edges of E are in $E(G)$ then*

$$\chi_E(G) = \sum_{E' \subseteq E : E' \neq E} \left(\sqrt{\frac{p}{1-p}}\right)^{|E|-|E'|} \chi_{E'}(G)$$

This simple observation whose proof is deferred to Section 6.4.4 can be applied recursively to replace a dense shape α by a sum of its sparse subshapes $\{\beta\}$.

Handling the subshapes $\{\beta\}$ requires some care. Destroying edges from a shape can cause its norm to either go up or down: the vertex separator gets smaller (increasing the norm), but if we remove edges from inside the SMVS, the norm goes down. An important observation is we do not necessarily have to apply Observation 6.16 on the entire set of edges of a shape, but we can also just apply it on some of the edges. We will choose a set of edges $\mathrm{Res}(\alpha) \subseteq E(\alpha)$ that "protects the minimum vertex separator" and only

[4] Beware two uses of the terms "dense"/"sparse". Here it refers to a small graph (the shape or a small subgraph) as opposed to the overall input $G \sim G_{n,p}$.

apply conditioning on edges outside $\mathrm{Res}(\alpha)$. In this way the norm of subshapes β will be guaranteed to be less than α. The fact that it's possible to reserve such edges is shown separately for the different kind of shapes we encounter in our analysis.

Method #2: Frobenius norm trick A relatively simple calculation shows that shapes with many edges have small Frobenius norm $\|M_\alpha\|_F$. The "tricky" part of the trick is that we can bound $\|M_\alpha\| \leq \|M_\alpha\|_F$ for *all* shapes simultaneously without using a union bound over α, by instead using Cauchy-Schwarz.

The Frobenius norm bound works well for shapes that have $\Omega(D_{\mathrm{SoS}})$ excess edges (edges beyond $|V(\alpha) \setminus (U_\alpha \cup V_\alpha)|$, which is what's needed for connectivity). It doesn't work well for the identity matrix, i.e. the trivial shape with $|U_\alpha| = |V_\alpha| = k$. For this shape, the Frobenius norm is $\binom{n}{k}$ whereas the true norm is just 1.

6.2.4 Informal sketch for bounding τ and τ_P

The most important part of the approximate PSD decomposition in Section 3.3 is showing that middle shapes τ and intersection terms τ_P can be charged to the identity matrix, i.e. a type of "graph matrix tail bound" for middle shapes and intersection terms. In this subsection, we describe the properties that τ and τ_P satisfy, their coefficients, and their norm bounds. Using this, we show that for each individual non-trivial τ, $\lambda_\tau \|M_\tau\| \ll 1$ and for each intersection pattern P, $\lambda_{\gamma \circ \tau \circ \gamma'^{\intercal}} \|M_{\tau_P}\| \ll 1$. The combinatorial arguments in this section crucially rely on the connected truncation, which enforces that all vertices in α have a path to $U_\alpha \cup V_\alpha$.

Middle shapes

The decomposition of a shape into left, middle, and right parts respects the connected truncation:

Proposition 6.17. *If all vertices in α have a path to $U_\alpha \cup V_\alpha$, then decomposing $\alpha = \sigma \circ \tau \circ \sigma'^{\intercal}$, all vertices in τ have a path to both U_τ and V_τ.*

Proof. It suffices to show that there is a path to $U_\tau \cup V_\tau$. In this case, say there is a path to vertex $u \in U_\tau$, then u must have a path to V_τ. Otherwise, $U_\tau \setminus \{u\}$ would be a smaller vertex separator of τ than U_τ, a contradiction to τ being a middle shape.

Let $v \in V(\tau) \subseteq V(\alpha)$. By assumption there is a path from v to $u \in U_\alpha \cup V_\alpha$; without loss of generality, $u \in U_\alpha$. Since v is not in σ, which was constructed by taking all vertices reachable from U_α without passing through U_τ, the path must pass through U_τ. □

Proposition 6.18. *For each middle shape τ such that $|V(\tau)| > \frac{|U_\tau|+|V_\tau|}{2}$ and every vertex in τ is connected to $U_\tau \cup V_\tau$, $\lambda_\tau \|M_\tau\| \ll 1$.*

Proof. The coefficient λ_τ is

$$\lambda_\tau = \left(\frac{k}{n}\right)^{|V(\tau)| - \frac{|U_\tau| + |V_\tau|}{2}} \left(-\sqrt{\frac{p}{1-p}}\right)^{|E(\tau)|}$$

The norm bound on M_τ is

$$\|M_\tau\| \leq \widetilde{O}\left(n^{\frac{|V(\tau)\setminus S|}{2}} \left(\sqrt{\frac{1-p}{p}}\right)^{|E(S)|}\right)$$

where S is the sparse minimum vertex separator of τ. We now have that $|\lambda_\tau| \|M_\tau\|$ is

$$\widetilde{O}\left(\left(\frac{k}{\sqrt{n}}\sqrt{\frac{p}{1-p}}\right)^{|V(\tau)| - \frac{|U_\tau| + |V_\tau|}{2}} \left(\sqrt{\frac{1-p}{np}}\right)^{|S| - \frac{|U_\tau| + |V_\tau|}{2}} \left(\sqrt{\frac{p}{1-p}}\right)^{|E(\tau)\setminus E(S)| - |V(\tau)\setminus S|}\right).$$

We claim that each of the three terms is upper bounded by 1. This will prove the claim, since the first term provides a decay for $|V(\tau)| > \frac{|U_\tau| + |V_\tau|}{2}$. The base of the first term is less than 1 for $k \lessapprox n/\sqrt{d}$. The exponent of the second term is nonnegative: since τ is a middle shape, both U_τ and V_τ are minimum vertex separators of τ, and since S is a vertex separator, it must have larger size. The exponent of the third term is nonnegative by the following lemma.

Lemma 6.19. *For a middle shape τ such that every vertex is connected to $U_\tau \cup V_\tau$, and any vertex separator S of τ,*

$$|E(\tau) \setminus E(S)| \geq |V(\tau) \setminus S|.$$

Proof. First, we claim that every vertex in τ is connected to S. Let $v \in V(\tau)$, and by the connected truncation, v is connected to some $u \in U_\tau$. Since τ is a middle shape, there must be a path from u to V_τ (else $U_\tau \setminus \{u\}$ would be a smaller vertex separator than U_τ). The path necessarily passes through S since S is a vertex separator, therefore we now have a path from v to S.

Now, we can assign an edge of $E(\tau) \setminus E(S)$ to each vertex of $V(\tau) \setminus S$ to prove the claim. To do this, run a breadth-first search from S, and assign an edge to the vertex that it explores. □

□

Intersection terms

Recall that intersection terms are formed by the intersection of $\sigma, \tau, \sigma'^{\intercal}$. Then we factor out the non-intersecting parts, leaving the *middle intersection* τ_P, which is an intersection of some portion γ of σ, the middle shape τ, and some portion γ' of σ', which we now make formal.

Definition 6.20 (Middle intersection). *Let γ, γ' be left shapes and τ be a shape such that $\gamma \circ \tau \circ \gamma'^{\intercal}$ are composable. We say that an intersection pattern $P \in \mathcal{P}_{\gamma,\tau,\gamma'^{\intercal}}$ is a middle intersection if U_γ is a minimum vertex separator in γ of U_γ and $V_\gamma \cup \mathrm{Int}(P)$. Similarly, $U_{\gamma'}$ is a minimum vertex separator in γ' of $U_{\gamma'}$ and $V_{\gamma'} \cup \mathrm{Int}(P)$. Finally, we also require that P has at least one intersection.*

Let $\mathcal{P}^{mid}_{\gamma,\tau,\gamma'}$ denote the set of middle intersections.

Remark 6.21. *For middle intersections we use the notation τ_P to denote the resulting shape, as compared to α_P which is used for an arbitrary intersection pattern.*

Remark 6.22. *In fact this definition also captures recursive intersection terms which are created from later rounds of factorization. We say that $P \in \mathcal{P}_{\gamma_k,\dots,\gamma_1,\tau,\gamma_1'^{\intercal},\dots,\gamma_k'^{\intercal}}$ is a middle intersection, denoted $P \in \mathcal{P}^{mid}_{\gamma_k,\dots\gamma_1,\tau,\gamma_1',\dots,\gamma_k'}$, if for all $j = 0, \dots, k-1$, letting τ_j be the shape of intersections so far between $\gamma_j, \dots, \gamma_1, \tau, \gamma_1'^{\intercal}, \dots, \gamma_j'^{\intercal}$, the intersection $\gamma_{j+1}, \tau_j, \gamma_{j+1}'^{\intercal}$ is a middle intersection.*

We need the following structural property of middle intersections.

Proposition 6.23. *For a middle intersection $P \in \mathcal{P}^{mid}_{\gamma,\tau,\gamma'}$ such that every vertex of τ is connected to both U_τ and V_τ, every vertex in τ_P is connected to both U_{τ_P} and V_{τ_P}.*

Proof. By assumption, all vertices in τ have a path to both U_τ and V_τ. Since $U_\tau = V_\gamma$, and all vertices in γ have a path to U_γ by definition of a left shape, all vertices in τ have a path to $U_\gamma = U_{\tau_P}$. Similarly, $V_\tau = U_{\gamma'}$ is connected to $V_{\gamma'} = V_{\tau_P}$. Thus we have shown that all vertices in τ are connected to both U_{τ_P} and V_{τ_P}.

Vertices in γ have a path to U_{τ_P} by definition; we must show that they also have a path to V_{τ_P}. A similar argument will hold for vertices in γ'. Since all vertices in γ have paths to U_γ, it suffices to show that all $u \in U_\gamma$ have a path to V_{τ_P}. There must be a path from u to some $v \in V_\gamma \cup \mathrm{Int}(P)$, else this would violate the definition of a middle intersection by taking $U_\gamma \setminus \{u\}$. The vertex v is in either τ or γ', both of which are connected to V_{τ_P}, hence we are done. □

Now we show that middle intersections have small norm. We focus on the first level of intersection terms; the general case of middle intersections in Remark 6.22 follows by induction.

Proposition 6.24. *For left shapes γ, γ', proper middle shape τ such that every vertex in τ is connected to $U_\tau \cup V_\tau$, and a middle intersection $P \in \mathcal{P}^{mid}_{\gamma,\tau,\gamma'}$, $\left|\lambda_{\gamma\circ\tau\circ\gamma'^{\intercal}}\right| \left\|M_{\tau_P}\right\| \ll 1$.*

Proof. For an intersection pattern P, the coefficient is

$$\lambda_{\gamma\circ\tau\circ\gamma'^{\intercal}} = \left(\frac{k}{n}\right)^{|V(\tau_P)|+i_P-\frac{|U_{\tau_P}|+|V_{\tau_P}|}{2}} \left(-\sqrt{\frac{p}{1-p}}\right)^{|E(\tau_P)|}$$

where $i_P := \left|V(\gamma \circ \tau \circ \gamma'^{\intercal})\right| - |V(\tau_P)|$ is the number of intersections in P.

The norm bound on M_{τ_P} is

$$\widetilde{O}\left(\max_{\beta,S}\left\{n^{\frac{|V(\beta)|+|I_\beta|-|S|}{2}}\left(\sqrt{\frac{1-p}{p}}\right)^{|E(S)|+|E(\tau_P)|-|E(\beta)|-2|E_{phantom}|}\right\}\right)$$

where the maximization is taken over all linearizations β of τ_P and all separators S for β. Here $E_{phantom}$ are the edges which are in τ_P but not in β.

We now have that $\left|\lambda_{\gamma\circ\tau\circ\gamma'^{\intercal}}\right|\left\|M_{\tau_P}\right\|$ is

$$\widetilde{O}\Bigg(\max_{\beta,S}\Bigg\{\left(\frac{k}{\sqrt{n}}\sqrt{\frac{p}{1-p}}\right)^{|V(\gamma\circ\tau\circ\gamma'^{\intercal})|-\frac{|U_\beta|+|V_\beta|}{2}}\left(\sqrt{\frac{1-p}{np}}\right)^{i_P-|I_\beta|+|S|-\frac{|U_\beta|+|V_\beta|}{2}}\cdot$$
$$\left(\sqrt{\frac{p}{1-p}}\right)^{|E(\beta)\setminus E(S)|-|V(\beta)\setminus S|-|I_\beta|+2|E_{phantom}|}\Bigg\}\Bigg).$$

We claim that each of the three terms is upper bounded by 1, which proves the proposition since the first term provides a decay (since an intersection is nontrivial). The base of the first term (vertex decay) is less than 1 for $k \lessapprox n/\sqrt{d}$. The second term has a nonnegative exponent by the *intersection tradeoff lemma* from [BHK+16]. The version we cite is Lemma 9.32 in [PR20], with the simplification that τ is a proper middle shape, so $I_\tau = \emptyset$ and $\left|S_{\tau,min}\right| = |U_\tau| = |V_\tau|$ (the full form is used for intersection terms with $\gamma_1, \ldots, \gamma_k, k > 1$).

Lemma 6.25 (Intersection tradeoff lemma)**.** *For all left shapes γ, γ' and proper middle shapes τ, let $P \in \mathcal{P}_{\gamma,\tau,\gamma'^{\intercal}}$ be a middle intersection, then*

$$|V(\tau)| - \frac{|U_\tau| + |V_\tau|}{2} + |V(\gamma)| - |U_\gamma| + |V(\gamma')| - |U_{\gamma'}| \geq |V(\tau_P)| + |I_{\tau_P,min}| - |S_{\tau_P,min}|$$

where $S_{\alpha,min}$ is defined to be a minimum vertex separator of α with all multi-edges deleted, and $I_{\alpha,min}$ is the set of isolated vertices in α with all multi-edges deleted.

The above inequality is equivalent to

$$i_P + \frac{|U_\tau| + |V_\tau|}{2} - |U_\gamma| - |U_{\gamma'}| \geq \left|I_{\tau_P,min}\right| - \left|S_{\tau_P,min}\right|.$$

Since $|U_\tau| \leq |U_\gamma| = |U_\beta|, |V_\tau| \leq |U_{\gamma'}| = |V_\beta|, \left|I_{\tau_P,min}\right| \geq |I_\beta|, |S| \geq \left|S_{\tau_P,min}\right|$, this implies

$$i_P - |I_\beta| + |S| - \frac{|U_\beta| + |V_\beta|}{2} \geq 0.$$

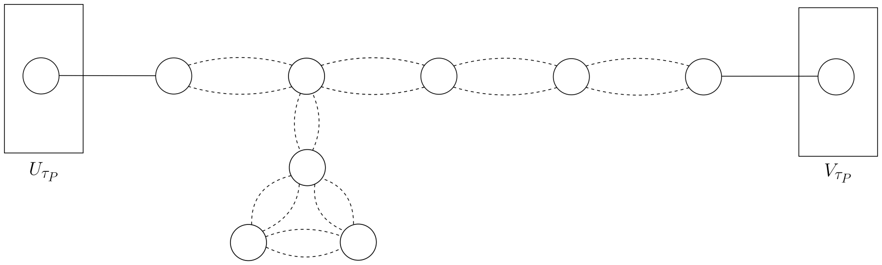


Figure 6.1: Dashed edges denote edges in $E_{phantom}$. The 6 vertices that are only incident to dashed edges are isolated in β. Note that $S = \emptyset$ is a separator for β.

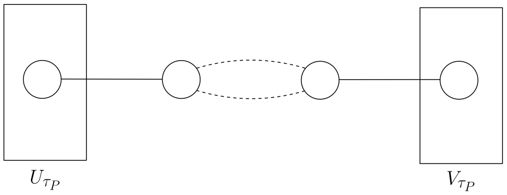


Figure 6.2: The shape from Fig. 6.1 after contracting edge pairs that have been charged to isolated vertices.

The third term has a nonnegative exponent, as the following lemma shows. The proof of this important lemma uses an explicit charging scheme; a second proof using induction is given in [JPR+21, Lemma 6.9].

Lemma 6.26. *For any linearization β of τ_P and all separators S for β,*

$$|E(\beta)| + 2|E_{phantom}| - |E(S)| \geq |V(\tau_P)| + |I_\beta| - |S|.$$

Proof. We first reduce $E_{phantom}$ to only the set of edges which fully disappeared from τ_P (each multiedge in τ_P was linearized to either a single edge or no edge; only keep the phantom edges that linearize to no edge). Furthermore, only keep two parallel multiedges even if more disappeared.

Let τ_P^{phant} be the multigraph formed from β plus the pairs of phantom multiedges in $E_{phantom}$. We need to assign each vertex of $V\left(\tau_P^{phant}\right) \setminus V(S)$ an edge of $E\left(\tau_P^{phant}\right) \setminus E(S)$, and we need to assign isolated vertices in I_β two edges. Note that the connectivity of τ_P^{phant} is exactly the same as τ_P. The difference between the two shapes is that the nonzero edge multiplicities are modified.

For vertices that are in the same connected component of τ_P^{phant} as a vertex in S, consider running a breadth-first search from S, and assign each edge to the vertex it explores. Vertices in I_β must be explored via a double edge, in order for them to become isolated during linearization, so assign both.

For vertices that are not in the same component of τ_P^{phant} as S, the edge assignment is more complicated. Let C be a component, for example as depicted in Fig. 6.1. Because of the connected truncation, specifically Proposition 6.23, component C must intersect both U_β and V_β. For the isolated vertices in $I_\beta \cap C$, order them by their distance from $U_\beta \cup V_\beta$. Charge them to the two edges along the shortest path to $U_\beta \cup V_\beta$. For the remaining non-isolated vertices, we claim that there is at least one additional double edge in C that has not yet been charged. As noted just above using Proposition 6.23, there is a path P between U_β and V_β in this component. However, since C does not intersect S, which is by assumption a separator in β, the path cannot be entirely in β i.e. it must contain at least one double edge. If P does not pass through an isolated vertex, this double edge evidently has not yet been charged. If P passes through an isolated vertex, because all edges incident to isolated vertices are double edges, there must be more double edges in P than isolated vertices in P. Since each isolated vertex only charges one incident double edge, they can't all be charged. In either case, P contains an uncharged double edge.

Now contract the edges in τ_P^{phant} that were charged for isolated vertices as in Fig. 6.2. Order the non-isolated vertices by their distance from this double edge. The double edge can be used to charge its two endpoints (which are not isolated), and the other vertices can be charged using the next edge in the shortest path to the double edge. This completes the charging, and the proof of the lemma. □

□

6.2.5 Proof of Theorem 6.7

The lemmas in the preceding section are informal, but they actually show something more that justifies Theorem 6.7. For technical reasons, our formal proof that implements the proof outline will break for $d \geq n^{0.5}$, and thus it doesn't cover Theorem 6.7, though we show that the norm bounds, charging arguments, plus the argument of [BHK+16] are already sufficient to prove Theorem 6.7.

In the proof of Proposition 6.18 the term $\lambda_\tau \|M_\tau\|$ was broken into three parts, each less than 1. Using just the first two parts,

$$|\lambda_\tau| \|M_\tau\| \leq \widetilde{O}\left(\left(\frac{k}{\sqrt{n}}\sqrt{\frac{p}{1-p}}\right)^{|V(\tau)\setminus S|}\left(\frac{k}{n}\right)^{|S|-\frac{|U_\tau|+|V_\tau|}{2}}\right).$$

Letting $k = \frac{n}{d^{1/2+\varepsilon}}$,

$$\begin{aligned}|\lambda_\tau| \|M_\tau\| &\leq \widetilde{O}\left(\left(\frac{1}{d^\varepsilon}\right)^{|V(\tau)|-|S|}\left(\frac{1}{d^{1/2+\varepsilon}}\right)^{|S|-\frac{|U_\tau|+|V_\tau|}{2}}\right) \\ &\leq \widetilde{O}\left(\left(\frac{1}{d^\varepsilon}\right)^{|V(\tau)|-|S_\tau|}\left(\frac{1}{d^{1/2+\varepsilon}}\right)^{|S_\tau|-\frac{|U_\tau|+|V_\tau|}{2}}\right)\end{aligned}$$

where S_τ is the dense MVS of τ. Observe that *this equals $\lambda_\tau \| M_\tau \|$ in the dense case* ($p = 1/2$) *for a random graph of size d.* That is, suppose we performed the pseudocalibration and norm bounds from [BHK+16] for a random graph $G \sim G_{d,1/2}$. Then we would get $\lambda_\tau = \left(\frac{1}{d^{1/2+\varepsilon}}\right)^{|V(\tau)| - \frac{|U_\tau|+|V_\tau|}{2}}$ and $\|M_\tau\| \leq (\log d)^{O(1)} \sqrt{d}^{|V(\tau)|-|S_\tau|}$, and $\lambda_\tau \|M_\tau\|$ is the same as the above. The main point is: by the analysis from [BHK+16], we know that for shapes up to size $\Omega(\log d)$, the sum of these norms is $o(1)$. Therefore[5] our matrices sum to $o(1)$ for SoS degree $\Omega(\log d)$.

The same phenomenon occurs for the intersection terms. Essentially we are neglecting the effect of the edges and considering only the vertex factors. By taking the first two out of three terms used in the proof of Proposition 6.24 we have the bound ($k = \frac{n}{d^{1/2+\varepsilon}}$):

$$\begin{aligned} \left|\lambda_{\gamma \circ \tau \circ \gamma'^\intercal}\right| \|M_{\tau_P}\| &\leq \widetilde{O}\left(\left(\frac{1}{d^\varepsilon}\right)^{|V(\tau_P)|+|I_\beta|-|S|} \left(\frac{1}{d^{1/2+\varepsilon}}\right)^{|V(\gamma\circ\tau\circ\gamma'^\intercal)|-|V(\tau_P)|-|I_\beta|+|S|-\frac{|U_\beta|+|V_\beta|}{2}}\right) \\ &\leq \widetilde{O}\left(\left(\frac{1}{d^\varepsilon}\right)^{|V(\tau_P)|+|I_{\tau_P,min}|-|S_{\tau_P,min}|} \left(\frac{1}{d^{1/2+\varepsilon}}\right)^{|V(\gamma\circ\tau\circ\gamma'^\intercal)|-|V(\tau_P)|-|I_{\tau_P,min}|+|S_{\tau_P,min}|-\frac{|U_{\tau_P}|+|V_{\tau_P}|}{2}}\right) \end{aligned}$$

where $S_{\tau_P,min}$ and $I_{\tau_P,min}$ are the dense MVS and isolated vertices respectively in the graph τ_P after deleting all multiedges. For $G \sim G_{d,1/2}$, the pseudocalibration coefficient and norm bounds are

$$\begin{aligned} \lambda_{\gamma\circ\tau\circ\gamma'^\intercal} &= \left(\frac{1}{d^{1/2+\varepsilon}}\right)^{|V(\gamma\circ\tau\circ\gamma'^\intercal)|-\frac{|U_\gamma|+|U_{\gamma'}|}{2}} \\ \|M_{\tau_P}\| &\leq (\log d)^{O(1)} \sqrt{d}^{|V(\tau_P)|-|S_{\tau_P,min}|+|I_{\tau_P,min}|}. \end{aligned}$$

Multiplying these together, one can see that $\lambda_{\gamma\circ\tau\circ\gamma'^\intercal} \|M_{\tau_P}\|$ is the same. Therefore, the analysis of [BHK+16] also shows that the sum of all intersection terms is negligible.

By passing to the "dense proxy graph" $G \sim G_{d,1/2}$, we can essentially use [BHK+16] to deduce a degree-$\Omega(\log d)$ lower bound in the sparse case just using connected truncation with the sparse norm bounds. The analysis of [BHK+16] is limited to SoS degree $\Omega(\log d)$ and $d \geq n^\varepsilon$. In order to push the SoS degree up and the graph degree down in the remaining sections, we need to utilize conditioning and handle the combinatorial terms more carefully.

[5] The log factor is $(\log n)^{O(1)}$ for our norm bound, vs $(\log d)^{O(1)}$ for $G \sim G_{d,1/2}$, but these are the same up to a constant for $d \geq n^\varepsilon$.

6.3 Pseudocalibration with connected truncation

6.3.1 *The failure of "Just try pseudocalibration"*

Despite the power of pseudocalibration in guiding the construction of a pseudomoment matrix in SoS lower bounds, it heavily relies upon a planted distribution that is hard to distinguish from the null distribution. Unfortunately, a "quiet" planted distribution remains on the search in our setting.

Towards this end, we will consider the following "naïve" planted distribution $\mathcal{D}_{pl}$ that likely would have been many peoples' first guess:

(1) Sample a random graph $G \sim G_{n,p}$;

(2) Sample a subset $S \subseteq [n]$ by picking each vertex with probability $\frac{k}{n}$;

(3) Let $\tilde{G}$ be G with edges inside S removed, and output $(S, \tilde{G})$.

Proposition 6.27. *For all $d = O(\sqrt{n})$ and $k = \Omega(n/d)$, the naïve planted distribution $\mathcal{D}_{pl}$ is distinguishable in polynomial time from $G_{n,p}$ with $\Omega(1)$ probability.*

Proof. The number of labeled 4-cycles in $G_{n,d/n}$ has expectation $\mathbb{E}[C_4] = \frac{d^4}{n^4} \cdot n(n-1)(n-2)(n-3)$ and variance $\operatorname{Var} C_4 = O(d^4)$. In $\mathcal{D}_{pl}$ the expected number of labeled 4-cycles is

$$\begin{aligned}\mathop{\mathbb{E}}_{pl}[C_4] &= \frac{d^4}{n^4} \cdot n(n-1)(n-2)(n-3) \cdot \left(\left(1-\frac{k}{n}\right)^4 + 4\frac{k}{n}\left(1-\frac{k}{n}\right)^3 + 2\left(\frac{k}{n}\right)^2\left(1-\frac{k}{n}\right)^2\right) \\ &= \frac{d^4}{n^4} \cdot n(n-1)(n-2)(n-3) \cdot \left(1 - O(k^2/n^2)\right) \\ &= \mathbb{E}[C_4] - O(d^4k^2/n^2) = \mathbb{E}[C_4] - O(d^2)\end{aligned}$$

Since this is less than $\mathbb{E}[C_4]$ by a factor on the order of $\sqrt{\operatorname{Var} C_4}$, counting 4-cycles succeeds with constant probability. □

Remark 6.28. *To beat distinguishers of this type, it may be possible to construct the planted distribution from a sparse quasirandom graph (in the sense that all small subgraph counts match $G_{n,p}$ to leading order) which has an independent set size of $\Omega(n/\sqrt{d})$. In the dense setting, the theory of quasirandom graphs states that if the planted distribution and $G_{n,1/2}$ match the subgraph count of C_4, this is sufficient for all subgraph counts to match; in the sparse setting this is no longer true and the situation is more complicated [CG02].*

Remark 6.29. *Coja-Oghlan and Efthymiou [COE15] show that a slight modification of this planted distribution (correct the expected number of edges to be $p\binom{n}{2}$) is indistinguishable from the random distribution provided k is slightly smaller than the size of the maximum independent set in $G_{n,p}$. This is not useful for pseudocalibration because we are trying to plant an independent set with larger-than-expected size.*

The Fourier coefficients of the planted distribution are:

Lemma 6.30. *Let $x_T(S)$ be the indicator function for T being in the planted solution i.e. $T \subseteq S$. Then, for all $T \subseteq [n]$ and $\alpha \subseteq \binom{[n]}{2}$,*

$$\mathop{\mathbb{E}}_{(S,\tilde{G})\sim\mathcal{D}_{pl}}[x_T(S)\cdot\chi_\alpha(\tilde{G})] = \left(\frac{k}{n}\right)^{|V(\alpha)\cup T|}\left(\frac{p}{1-p}\right)^{\frac{|E(\alpha)|}{2}}$$

Proof. First observe that if any vertex of $V(\alpha)\cup T$ is outside S, then the expectation is 0. This is because either T is outside S, in which case $x_T = 0$, or a vertex of α is outside S, in which case the expectation of any edge incident on this vertex is 0 so the entire expectation is 0 using independence. Now, each vertex of $V(\alpha)\cup T$ is in S independently with probability $\left(\frac{k}{n}\right)^{|V(\alpha)\cup T|}$. Conditioned on this event happening, the character is simply $\chi_\alpha(0) = \left(\frac{p}{1-p}\right)^{\frac{|E(\alpha)|}{2}}$. Putting them together gives the result. □

This planted distribution motivates the following incorrect definition of the pseudo-expectation operator.

Definition 6.31 (Incorrect definition of $\widetilde{\mathbb{E}}$)**.** *For $S \subseteq [n], |S| \le D_{\text{SoS}}$,*

$$\widetilde{\mathbb{E}}[x^S] := \sum_{\substack{\alpha\subseteq\binom{[n]}{2}:\\ |\alpha|\le D_V}} \left(\frac{k}{n}\right)^{|V(\alpha)\cup S|}\left(\frac{p}{1-p}\right)^{\frac{|E(\alpha)|}{2}}\chi_\alpha(G)$$

For this operator, $\widetilde{\mathbb{E}}[1] = 1+\Omega(1)$. More generally, tail bounds of graph matrix sums are not small, which ruins our analysis technique.

6.3.2 *Salvaging the doomed*

We will now discuss our novel truncation heuristic. The next paragraphs are for discussion and the technical proof resumes at Definition 6.33. Letting $\mathcal{S}$ be the set of shapes α that contribute to the moment $\widetilde{\mathbb{E}}[x^I]$, the "connected truncation" restricts $\mathcal{S}$ to shapes α such that all vertices in α have a path to I (or in the shape view, to $U_\alpha \cup V_\alpha$).

Why might this be a good idea? Consider the planted distribution. The only tests we know of to distinguish the random/planted distributions are counting the number of edges or counting occurrences of small subgraphs. These tests cannot be implemented using only connected Fourier characters; shapes with disconnected middle vertices are needed to count small subgraphs. For example, suppose we fix a particular vertex $v \in V(G)$, and we consider the set of functions on G which are allowed to depend on v but are otherwise symmetric in the vertices of G. A basis for these functions can be made by taking shapes α such that U_α has a single vertex, $V_\alpha = \emptyset$, then fixing $U_\alpha = v$ (i.e. take the vector entry in row v). Let sym_v denote this set of functions.

Proposition 6.32. *Let $T(G)$ be the triangle counting function on G. Let $\mathrm{conn}(\mathsf{sym}_v)$ be the subset of sym_v such that all vertices have a path to v. Then $T(G) \notin \mathrm{span}(\mathrm{conn}(\mathsf{sym}_v))$.*

Proof. $T(G)$ has a unique representation in sym_v which requires disconnected Fourier characters. For example, one component of the function is $T_{\overline{v}}(G) =$ number of triangles not containing v. This is three vertices x, y, z outside of U_α with $\mathbb{1}_{(x,y)\in E(G)}\mathbb{1}_{(x,z)\in E(G)}\mathbb{1}_{(y,z)\in E(G)}$. The edge indicator function is implemented by Proposition 6.10. But there are no edges between x, y, z and $U_\alpha = v$ in these shapes.

It's not even possible to implement $T_v(G) =$ number of triangles containing v, as a required shape is two vertices x, y outside of U_α connected with an edge. □

Despite the truncation above, our pseudocalibration operator is not a "local function" of the graph. Our $\widetilde{\mathbb{E}}[x^S]$ can depend on vertices that are far away from S, but in an attenuated way. The graph matrix for α is a sum of all ways of overlaying the vertices of α onto G. The edges do not need to be overlaid. If an edge "misses" in G, then we can use this edge to get far away from S, but we take a decay factor of $\chi_e(0) = \sqrt{\frac{p}{1-p}} = \sqrt{\frac{d}{n-d}}$.

The "local function" property of $\widetilde{\mathbb{E}}[x^S]$ is also a connected truncation, but it is a connected truncation in a different basis. The basis is the 0/1 basis of $1_H(G)$ for $H \subseteq \binom{n}{2}$. A reasonable definition of graph matrices in this basis is

$$M_\alpha = \sum_{\text{injective } \sigma: V(\alpha)\to[n]} 1_{\sigma(E(\alpha))}(G)$$

which sums all ways to embed α into G. $\widetilde{\mathbb{E}}[x^S]$ is a local function if and only if in this basis it is a sum of shapes α satisfying the (same) condition that all vertices are connected to $S = U_\alpha \cup V_\alpha$.

For sparse graphs, the two bases are somewhat heuristically interchangeable since:

$$\frac{1}{p}1_e(G) = 1 - \sqrt{\frac{1-p}{p}}\chi_e(G) \approx -\sqrt{\frac{1-p}{p}}\chi_e(G).$$

Comparing the 0/1 basis and the Fourier basis, the 0/1 basis expresses combinatorial properties such as subgraph counts more nicely, while spectral analysis is only feasible in the Fourier basis. In the proof (see Definition 6.56), we will augment ribbons so that they may also contain 0/1 indicators, and this flexibility helps us overcome both the spectral and combinatorial difficulties in the analysis.

Using connected objects to take advantage of correlation decay is also a theme in the cluster expansion from statistical physics (see Chapter 5 of [FV18]). Although not formally connected with connected truncation, the two methods share some similar characteristics.

Recall that we are given a graph $G \sim G_{n,p}$ where $d = pn$ is the average degree and our goal is to show that whp, degree D_{SoS} SoS thinks there exists an independent set of size $k = \frac{n}{d^{1/2} D_{\text{SoS}}^{c_0} \log n}$ for some absolute constant c_0, whereas the true independent set has size $\approx \frac{n \log d}{d}$. Formally we define the candidate moment matrix as:

Definition 6.33 (Moment matrix)**.**

$$\Lambda := \sum_{\alpha \in \mathcal{S}} \left(\frac{k}{n}\right)^{|V(\alpha)|} \cdot \left(\frac{p}{1-p}\right)^{\frac{|E(\alpha)|}{2}} \frac{M_\alpha}{|\mathrm{Aut}(\alpha)|}.$$

where $\mathcal{S}$ is the set of proper shapes such that

1. $|U_\alpha|, |V_\alpha| \leq D_{\text{SoS}}$.
2. $|V(\alpha)| \leq D_V$ *where* $D_V = C D_{\text{SoS}} \log n$ *for a sufficiently large constant* C.
3. *Every vertex in* α *has a path to* $U_\alpha \cup V_\alpha$.

We refer to $\left(\frac{k}{n}\right)^{|V(\alpha)|}$ as the "vertex decay factor" and $\left(\frac{p}{1-p}\right)^{\frac{|E(\alpha)|}{2}}$ as the "edge decay factor". The candidate moment matrix is the principal submatrix indexed by $\binom{[n]}{\leq D_{\text{SoS}}/2}$. The non-PSDness properties of Λ are easy to verify:

Lemma 6.34. Λ *is SoS-symmetric.*

Proof. This is equivalent to the fact that the coefficient of α does not depend on how $U_\alpha \cup V_\alpha$ is partitioned into U_α and V_α for $|U_\alpha \cup V_\alpha| \leq D_{\text{SoS}}$. □

Lemma 6.35. $\widetilde{\mathbb{E}}[1] = 1$.

Proof. For $U_\alpha = V_\alpha = \emptyset$, the only shape in which all vertices are connected to $U_\alpha \cup V_\alpha$ is the empty shape. Therefore $\widetilde{\mathbb{E}}[1] = 1$. □

Lemma 6.36. $\widetilde{\mathbb{E}}$ *satisfies the feasibility constraints.*

Proof. We must show that $\widetilde{\mathbb{E}}[x^S] = 0$ whenever S is not an independent set in G. Observe that if ribbon R contributes to $\widetilde{\mathbb{E}}[x^S]$, then if we modify the set of edges inside S, the resulting ribbon still contributes to $\widetilde{\mathbb{E}}[x^S]$. In fact, each edge also comes with a factor of $\sqrt{\frac{p}{1-p}}$. By Proposition 6.10, we can group these ribbons into an indicator function $\frac{1}{(1-p)^{\binom{|S|}{2}}} 1_{\overline{E(S)}}$. That is, $\widetilde{\mathbb{E}}[x^S] = 0$ if S has an edge. □

Lemma 6.37. *With probability at least* $1 - o_n(1)$, $\widetilde{\mathbb{E}}$ *has objective value* $\widetilde{\mathbb{E}}[\sum x_i] \geq (1 - o(1))k$.

Proof. Let $1_{(k)}$ denote the all-1s vector of length k.

$$\begin{aligned}\widetilde{\mathbb{E}}[\sum_i x_i] &= 1_{(n)}^{\mathsf{T}} \left(\sum_{\substack{\alpha\in\mathcal{S}:\\ |U_\alpha|=1, V_\alpha=\emptyset}} \left(\frac{k}{n}\right)^{|V(\alpha)|} \cdot \left(\frac{p}{1-p}\right)^{\frac{|E(\alpha)|}{2}} \frac{M_\alpha}{|\operatorname{Aut}(\alpha)|} \right) 1_{(1)} \\ &= \frac{k}{n} \cdot 1_{(n)}^{\mathsf{T}} \left(\sum_{\substack{\alpha\in\mathcal{S}:\\ |U_\alpha|=1, V_\alpha=\emptyset}} \left(\frac{k}{n}\right)^{|V(\alpha)| - \frac{|U_\alpha|+|V_\alpha|}{2}} \cdot \left(\frac{p}{1-p}\right)^{\frac{|E(\alpha)|}{2}} \frac{M_\alpha}{|\operatorname{Aut}(\alpha)|} \right) 1_{(1)} .\end{aligned}$$

The main term is the shape with a single vertex. This shape contributes k to the overall sum. Using the main lemmas to be proven later (Section 6.4.7), the remaining terms have total spectral norm bounded by $O(1)$ whp. Therefore, the remaining terms are bounded by

$$\frac{k}{n} \cdot O(1) \cdot \left\|1_{(n)}\right\| \left\|1_{(1)}\right\| = O(k/\sqrt{n}) .$$

□

What remains is to show $\Lambda \succeq 0$. To do this it's helpful to renormalize the matrix entries by multiplying the degree-(k,l) block by a certain factor.

Definition 6.38 (Shape coefficient)**.** *For all shapes α, let*

$$\lambda_\alpha := \left(\frac{k}{n}\right)^{|V(\alpha)| - \frac{|U_\alpha|+|V_\alpha|}{2}} \cdot \left(\frac{p}{1-p}\right)^{\frac{|E(\alpha)|}{2}} .$$

It suffices to show $\sum_{\alpha\in\mathcal{S}} \lambda_\alpha \frac{M_\alpha}{|\operatorname{Aut}(\alpha)|} \succeq 0$ because left and right multiplying by a rank-1 matrix and its transpose returns Λ.

Proposition 6.39. *If α, β are composable shapes, then $\lambda_\alpha \lambda_\beta = \lambda_{\alpha\circ\beta}$.*

6.4 PSD-ness

We now work towards the formal proof of Theorem 6.3. Recall that our goal is to show PSD-ness of the matrix $\sum_{\alpha\in\mathcal{S}} \lambda_\alpha \frac{M_\alpha}{|\operatorname{Aut}(\alpha)|}$.

Definition 6.40 (Properly composable)**.** *Composable ribbons $R_1, \ldots, R_k$ are properly composable if there are no intersections beyond the necessary ones $B_{R_i} = A_{R_{i+1}}$.*

Definition 6.41 (Proper composition)**.** *Given composable ribbons $R_1, \ldots, R_k$ which are not necessarily properly composable, with shapes $\alpha_1, \ldots, \alpha_k$, let $R_1 \odot R_2 \odot \cdots \odot R_k$ be the shape of $\alpha_1 \circ \cdots \circ \alpha_k$.*

That is, $R_1 \odot \cdots \odot R_k$ is the shape obtained by concatenating the ribbons but not collapsing vertices which repeat between ribbons, as if they were properly composable. Compare this with $R_1 \circ \cdots \circ R_k$, in which repetitions collapse.

Definition 6.42 ($\mathcal{L}$ and $\mathcal{M}$)**.** *Let $\mathcal{L}$ and $\mathcal{M}$ be the set of left and middle shapes in $\mathcal{S}$. Shapes in these sets will be denoted $\sigma, \gamma \in \mathcal{L}$ and $\tau \in \mathcal{M}$. We abuse notation and let $L, G \in \mathcal{L}, T \in \mathcal{M}$ denote ribbons of shapes in $\mathcal{L}, \mathcal{M}$ (following the convention of using Greek letters for shapes and Latin letters for ribbons).*

Formalizing the process described in Section 3.3, we have the following lemma. It is easier to formally manipulate unsymmetrized objects, ribbons, for the PSD decomposition and switch to symmetrized objects, shapes, only when we need to invoke norm bounds. The formal details are carried out in the next subsection.

Lemma 6.43. *(Decomposition in terms of ribbons).*

$$\sum_{\alpha \in \mathcal{S}} \lambda_\alpha \frac{M_\alpha}{|\mathrm{Aut}(\alpha)|} = \sum_{R \in \mathcal{S}} \lambda_R M_R =$$

$$\left(\sum_{L \in \mathcal{L}} \lambda_L M_L\right) \left(\sum_{j=0}^{2D_{\mathrm{SoS}}} (-1)^j \sum_{\substack{G_j,\dots,G_1,\\ T,\\ G'_1,\dots,G'_j}} \lambda_{G_j \odot \cdots \odot G_1 \odot T \odot {G'_1}^{\intercal} \odot \cdots \odot {G'_j}^{\intercal}} M_{G_j \circ \cdots \circ G_1 \circ T \circ {G'_1}^{\intercal} \circ \cdots \circ {G'_j}^{\intercal}} \right) \left(\sum_{L \in \mathcal{L}} \lambda_L M_L\right)^{\intercal}$$

$$+ \textit{ truncation error}_{\textit{too many vertices}} + \textit{truncation error}_{\textit{too many edges in one part}}$$

where

1.

$$\textit{truncation error}_{\textit{too many vertices}} = - \sum_{\substack{L,T,L':\\ |V(L \circ T \circ L'^{\intercal})| > D_V,\\ L,T,L'^{\intercal} \text{ are properly composable}}} \lambda_{L \circ T \circ L'^{\intercal}} M_{L \circ T \circ L'^{\intercal}}$$

$$+ \sum_{j=1}^{2D_{\mathrm{SoS}}} (-1)^{j+1} \sum_{\substack{L,G_j,\dots,G_1,T,G'_1,\dots,G'_j,L':\\ |V(L \odot G_j \odot \dots \odot G_1)| > D_V \text{ or}\\ |V({G'_1}^{\intercal} \odot \dots \odot {G'_j}^{\intercal} \odot L'^{\intercal})| > D_V}} \lambda_{L \odot G_j \odot \cdots \odot G_1 \odot T \odot {G'_1}^{\intercal} \odot \cdots \odot {G'_j}^{\intercal} \odot L'^{\intercal}} M_L M_{G_j \circ \cdots \circ G_1 \circ T \circ {G'_1}^{\intercal} \circ \cdots \circ {G'_j}^{\intercal}} M_{L'}^T$$

$$\textit{truncation error}_{\textit{too many edges in one part}} = \sum_{\substack{L,T,L': \\ |V(L\circ T\circ L'^{\intercal})|\le D_V, \\ |E_{mid}(T)|-|V(T)|>CD_{\mathrm{SoS}} \\ L,T,L'^{\intercal} \textit{ are properly composable}}} \lambda_{L\odot T\odot L'^{\intercal}} M_{L\circ T\circ L'^{T}}$$

$$+ \sum_{j=1}^{2D_{\mathrm{SoS}}} (-1)^j \sum_{\substack{L,G_j,\ldots,G_1,T,G'_1,\ldots,G'_j,L': \\ |V(L\odot G_j\odot\ldots\odot G_1)|\le D_V \textit{ and } |V(G_1'^{\intercal}\odot\ldots\odot G_j'^{\intercal}\odot L'^{\intercal})|\le D_V \\ |E_{mid}(G_j)|-|V(G_j)|>CD_{\mathrm{SoS}} \textit{ or } |E_{mid}(G'_j)|-|V(G'_j)|>CD_{\mathrm{SoS}} \\ L,(G_j\circ\ldots\circ G_1\circ T\circ G_1'^{\intercal}\circ\ldots\circ G_j'^{\intercal}),L'^{\intercal} \textit{ are properly composable}}} \lambda_{L\odot G_j\odot\ldots\odot G_j'^{\intercal}\odot L'^{\intercal}} M_{L\circ G_j\circ\ldots\circ G_j'^{\intercal}\circ L'^{T}}$$

where for a ribbon R, $E_{mid}(R)$ is the set of edges of R which are not contained in A_R, not contained in B_R, and are not incident to any vertices in $A_R \cap B_R$.

2. *in all of these sums, the ribbons $L, G_j, \ldots, G_1, T, G'_1, \ldots, G'_j, L'$ satisfy the following conditions:*

   (a) *$T \in \mathcal{M}$, $L, L' \in \mathcal{L}$, and each $G_i, G'_i \in \mathcal{L}$.*

   (b) *$L, G_j, \ldots, G_1, T, G_1'^{\intercal}, \ldots, G_j'^{\intercal}, L'^{\intercal}$ are composable.*

   (c) *The intersection pattern induced by $G_j, \ldots, G_1, T, G'_1, \ldots, G'_j$ is a middle intersection pattern.*

   (d) *$|V(T)| \le D_V$, $|V(G_j \odot \ldots \odot G_1)| \le D_V$, and $|V(G_1'^{\intercal} \odot \ldots \odot G_j'^{\intercal})| \le D_V$*

   (e) *Except when noted otherwise (which only happens for truncation error$_{\textit{too many edges in one part}}$), all of the ribbons $G_j, \ldots, G_1, T, G'_1, \ldots, G'_j$ (but not necessarily L, L') satisfy the constraint that $|E_{mid}(R)| - |V(R)| \le CD_{\mathrm{SoS}}$.*

6.4.1 *Proof of Lemma 6.43: formalizing the PSD Decomposition*

Here we prove Lemma 6.43 by formally going through the approximate PSD decomposition described in Section 3.3.

Proof. We abuse notation and write the sum over ribbons in $\mathcal{S}$ as

$$\sum_{\alpha\in\mathcal{S}} \lambda_\alpha \frac{M_\alpha}{|\mathrm{Aut}(\alpha)|} = \sum_{R\in\mathcal{S}} \lambda_R M_R.$$

(The automorphism group disappears as it was only there to ensure each ribbon is represented once anyway.)

From Proposition 2.45, every ribbon $R \in \mathcal{S}$ decomposes into $R = L \circ T \circ L'^{\intercal}$ where

$L, L' \in \mathcal{L}$ and $T \in \mathcal{M}$. We would like that

$$\sum_{R \in \mathcal{S}} \lambda_R M_R = \sum_{\substack{L, L' \in \mathcal{L}, T \in \mathcal{M}: \\ |V(L \circ T \circ L'^{\intercal})| \le D_V}} \lambda_{L \circ T \circ L'^{\intercal}} M_{L \circ T \circ L'^{\intercal}}.$$

However, this is not quite correct, as the ribbons on the right hand side may intersect.

Proposition 6.44.

$$\sum_{R \in \mathcal{S}} \lambda_R M_R = \sum_{\substack{L, L' \in \mathcal{L}, T \in \mathcal{M}: \\ |V(L \circ T \circ L'^{\intercal})| \le D_V, \\ L, T, L'^{\intercal} \textit{ are properly composable}}} \lambda_{L \circ T \circ L'^{\intercal}} M_{L \circ T \circ L'^{\intercal}}.$$

To start, we uncorrelate the sizes of L, T, L'. Add to the moment matrix

$$\sum_{\substack{L, L' \in \mathcal{L}, T \in \mathcal{M}: \\ |V(L \circ T \circ L'^{\intercal})| > D_V, \\ L, T, L'^{\intercal} \text{ are properly composable}}} \lambda_{L \circ T \circ L'^{\intercal}} M_{L \circ T \circ L'^{\intercal}}.$$

This term is added to truncation $\text{error}_{\text{too many vertices}}$.

We define matrices for intersection terms I_k, factored terms F_k, and truncation error T_k at level k recursively via the following process. We will maintain that the moment matrix satisfies for all k,

$$\sum_{R \in \mathcal{S}} \lambda_R M_R = \left(\sum_{L \in \mathcal{L}} \lambda_L M_L \right) \left(\sum_{i=1}^{k} (-1)^{i+1} F_i \right) \left(\sum_{L \in \mathcal{L}} \lambda_L M_L \right)^{\intercal} + (-1)^k I_k + \sum_{i=1}^{k} (-1)^i T_i.$$

In order to make this equation hold, given I_k we will choose F_{k+1}, T_{k+1}, and I_{k+1} so that

$$I_k = \left(\sum_{L \in \mathcal{L}} \lambda_L M_L \right) F_{k+1} \left(\sum_{L \in \mathcal{L}} \lambda_L M_L \right)^{\intercal} - T_{k+1} - I_{k+1}$$

At the start of the $(k+1)$th iteration, we will have that I_k is equal to a sum over middle ribbons T, "middle intersecting ribbons" $(G_k, \dots, G_1, G_1'^{\intercal}, \dots, G_k'^{\intercal}) \in \text{Mid}_T^{(k)}$, and non-intersecting ribbons L, L'. Furthermore, the ribbons $G_k, \dots, T, \dots, G_k'$ will satisfy $|E_{mid}(R)| - |V(R)| \le CD_{\text{SoS}}$, in which case we say that "edge bounds hold". Any time that a violation of this bound appears, we will throw the term into the truncation error for too many edges in one part. For example, initially we throw away T with too many edges. Initially,

$$\begin{aligned}
I_0 &:= \sum_{\substack{L,L'\in\mathcal{L},T\in\mathcal{M}:\\ |V(L)|\le D_V,|V(T)|\le D_V,\left|V(L'^{\intercal})\right|\le D_V,\\ L,T,L'^{\intercal}\text{ are properly composable,}\\ \text{edge bounds hold}}} \lambda_{L\odot T\odot L'^{\intercal}} M_{L\circ T\circ L'^{\intercal}}\\
T_0 &:= \sum_{\substack{L,L'\in\mathcal{L},T\in\mathcal{M}:\\ |V(L)|\le D_V,|V(T)|\le D_V,\left|V(L'^{\intercal})\right|\le D_V,\\ |E_{mid}(T)|-|V(T)|>CD_{\text{SoS}},\\ L,T,L'^{\intercal}\text{ are properly composable}}} \lambda_{L\odot T\odot L'^{\intercal}} M_{L\circ T\circ L'^{\intercal}}\\
F_0 &:= 0
\end{aligned}$$

Definition 6.45. *For a middle ribbon T, let $Mid_T^{(k)}$ be the collection of tuples of ribbons $G_k, \ldots, G_1$, $G_1', \ldots, G_k'$ such that the ribbons are composable and the induced intersection pattern of $G_k \circ \cdots \circ G_1 \circ T \circ G_1'^{\intercal} \circ \cdots \circ G_k'^{\intercal}$ is a middle intersection pattern.*

The inductive hypothesis for I_k is,

$$I_k = \sum_{\substack{T\in\mathcal{M},(G_k,\ldots,G_k')\in\text{Mid}_T^{(k)},L,L'\in\mathcal{L}:\\ \left|V(L\odot G_k\odot\ldots\odot G_1)\right|\le D_V,|V(T)|\le D_V,\left|V(G_1'^{\intercal}\odot\ldots\odot G_k'^{\intercal}\odot L'^{\intercal})\right|\le D_V,\\ L,G_k\circ\ldots\circ G_1\circ T\circ G_1'^{\intercal}\circ\ldots\circ G_k'^{\intercal},L'^{\intercal}\text{ are properly composable,}\\ \text{edge bounds satisfied}}} \lambda_{L\odot G_k\odot\ldots\odot G_1\odot T\odot G_1'^{\intercal}\odot\ldots\odot G_k'^{\intercal}\odot L'^{\intercal}} M_{L\circ G_k\circ\ldots\circ G_1\circ T\circ G_1'^{\intercal}\circ\ldots\circ G_k'^{\intercal}\circ L'^{T}}.$$

We can approximate I_k by

$$\left(\sum_{L\in\mathcal{L}}\lambda_L M_L\right) F_{k+1} \left(\sum_{L\in\mathcal{L}}\lambda_L M_L\right)^{\intercal}$$

where

$$F_{k+1} = \sum_{\substack{T\in\mathcal{M},(G_k,\ldots,G_k')\in\text{Mid}_T^{(k)}:\\ \left|V(G_k\odot\ldots\odot G_1)\right|\le D_V,|V(T)|\le D_V,\left|V(G_1'^{\intercal}\odot\ldots\odot G_k'^{\intercal})\right|\le D_V,\\ \text{edge bounds satisfied}}} \lambda_{G_k\odot\ldots\odot G_1\odot T\odot G_1'^{\intercal}\odot\ldots\odot G_k'^{\intercal}} M_{G_k\circ\ldots\circ G_1\circ T\circ G_1'^{\intercal}\circ\ldots\circ G_k'^{\intercal}}$$

In order to make it so that $I_k = \left(\sum_{L\in\mathcal{L}}\lambda_L M_L\right) F_{k+1} \left(\sum_{L\in\mathcal{L}}\lambda_L M_L\right)^{\intercal} - T_{k+1} - I_{k+1}$, we need to handle the following issues.

1. I_k only contains terms where $\left|V(L \odot G_k \odot \ldots \odot G_1)\right| \le D_V$ and $\left|V(G_1'^{\intercal} \odot \ldots \odot G_k'^{\intercal} \odot L'^{\intercal})\right| \le$

D_V, whereas some larger terms appear in the approximation. To handle this, we add

$$\sum_{\substack{T\in\mathcal{M},(G_k,\dots,G'_k)\in\mathrm{Mid}_T^{(k)},L,L'\in\mathcal{L}:\\ \left|V(G_k\odot\dots\odot G_1)\right|\le D_V,|V(T)|\le D_V,\left|V({G'_1}^\intercal\odot\dots\odot {G'_k}^\intercal)\right|\le D_V,\\ \left|V(L\odot G_k\odot\dots\odot G_1)\right|>D_V \text{ or } \left|V({G'_1}^\intercal\odot\dots\odot {G'_k}^\intercal\odot {L'}^\intercal)\right|>D_V,\\ \text{edge bounds satisfied}}} \lambda_{L\odot G_k\odot\dots\odot G_1\odot T\odot {G'_1}^\intercal\odot\dots\odot {G'_k}^\intercal\odot {L'}^\intercal} M_L M_{G_k\circ\dots\circ G_1\circ T\circ {G'_1}^\intercal\circ\dots\circ {G'_k}^\intercal} {M_{L'}}^\intercal$$

to T_{k+1}. This is the source of the truncation error terms with too many vertices.

2. I_k only contains terms where $L, G_k \circ \ldots \circ G_1 \circ T \circ {G'_1}^\intercal \circ \ldots \circ {G'_k}^\intercal, {L'}^\intercal$ are properly composable. The remaining terms are $L, G_k \circ \ldots \circ G_1 \circ T \circ {G'_1}^\intercal \circ \ldots \circ {G'_k}^\intercal, {L'}^\intercal$ such that they are not properly composable, and these will mostly be put into I_{k+1}.

$$\sum_{\substack{T\in\mathcal{M},(G_k,\dots,G'_k)\in\mathrm{Mid}_T^{(k)},L,L'\in\mathcal{L}:\\ \left|V(L\odot G_k\odot\dots\odot G_1)\right|\le D_V,|V(T)|\le D_V,\left|V({G'_1}^\intercal\odot\dots\odot {G'_k}^\intercal\odot {L'}^\intercal)\right|\le D_V,\\ L,G_k\circ\cdots G_1\circ T\circ {G'_1}^\intercal\circ\cdots {G'_k}^\intercal,{L'}^\intercal \text{ are not properly composable,}\\ \text{edge bounds satisfied}}} \lambda_{L\odot G_k\odot\dots\odot {G'_k}^\intercal\odot {L'}^\intercal} M_L M_{G_k\circ\cdots\circ {G'_k}^\intercal} {M_{L'}}^\intercal$$

These terms do not yet match our inductive hypothesis for I_{k+1}; for each $L, G_k \circ \ldots \circ G_1 \circ T \circ {G'_1}^\intercal \circ \ldots \circ {G'_k}^\intercal, {L'}^\intercal$ which are not properly composable, we must separate out the intersecting portions G_{k+1}, G'_{k+1}. To do this, decompose L as $L_2 \circ G_{k+1}$ where $B_{L_2} = A_{G_{k+1}}$ is the leftmost minimum vertex separator between A_L and $B_L \cup \{\text{intersected vertices}\}$. We decompose L' as $L'_2 \circ G'_{k+1}$ in a similar way. This is exactly the definition that $G_{k+1}, \ldots, G_1, T, G'_1, \ldots, G_{k+1}$ are a middle intersection, Definition 6.20. That is, $(G_{k+1}, \ldots, G'_{k+1}) \in \mathrm{Mid}^{(k+1)}$.

Claim 6.46. *L_2 and L'_2 are left ribbons.*

Proof. Definitionally, B_{L_2} is the unique minimum vertex separator of L_2. Furthermore, reachability of all vertices in L_2 is inherited from L. □

We record a lemma that will be needed later:

Lemma 6.47. *Factoring $L = L_2 \circ G_{k+1}$ is oblivious to edges inside A_L, B_L, edges incident to $A_L \cap B_L$, or edges not in L.*

Proof. Edges inside A_L or B_L do not affect the connectivity between A_L, B_L; the same is true for edges incident to $A_L \cap B_L$ since all vertices of $A_L \cap B_L$ must be taken in any separator of A_L and B_L. The last claim is clear because the factoring depends only on which vertices of L intersected, as well as the structure of L. □

Observe that now instead of summing over $L, \mathrm{Mid}^{(k)}, L'$, we may sum over $L_2, \mathrm{Mid}^{(k+1)}, L'_2$. That is, if we fix T and the ribbons $G_k, \ldots, G'_k$, then L determines the pair L_2, G_{k+1}, and vice versa.

We take I_{k+1} to be the subset of G_{k+1} that satisfy the edge bound $|E_{mid}(G_{k+1})| - |V(G_{k+1})| \leq CD_{\mathrm{SoS}}$. ($L_2$ and L'_2 are written as L, L' here so as to more clearly match the inductive hypothesis.)

$$\sum_{\substack{T \in \mathcal{M}, (G_{k+1}, \ldots, G'_{k+1}) \in \mathrm{Mid}_T^{(k+1)}, L, L' \in \mathcal{L}: \\ \left|V(L \odot G_{k+1} \odot \ldots \odot G_1)\right| \leq D_V, |V(T)| \leq D_V, \left|V({G'_1}^{\intercal} \odot \ldots \odot {G'_{k+1}}^{\intercal} \odot {L'}^{\intercal})\right| \leq D_V, \\ L, G_{k+1} \circ \ldots \circ G_1 \circ T \circ {G'_1}^{\intercal} \circ \ldots \circ {G'_{k+1}}^{\intercal}, {L'}^{\intercal} \text{ are properly composable,} \\ \text{edge bounds satisfied}}} \lambda_{L \odot G_{k+1} \odot \ldots \odot {G'_{k+1}}^{\intercal} \odot {L'}^{\intercal}} M_{L \circ G_{k+1} \circ \ldots \circ {G'_{k+1}}^{\intercal} \circ {L'}^{T}}$$

3. For technical reasons, we want to stop this process if $|E_{mid}(G_{k+1})| - |V(G_{k+1})| > CD_{\mathrm{SoS}}$ or $|E_{mid}(G'_{k+1})| - |V(G'_{k+1})| > CD_{\mathrm{SoS}}$. Thus we take such terms G_{k+1} and ${G'_{k+1}}^{\intercal}$ and add them to T_{k+1} instead of to I_{k+1}.

$$\sum_{\substack{T \in \mathcal{M}, (G_{k+1}, \ldots, G'_{k+1}) \in \mathrm{Mid}_T^{(k+1)}, L, L' \in \mathcal{L}: \\ \left|V(L \odot G_k \odot \ldots \odot G_1)\right| \leq D_V, |V(T)| \leq D_V, \left|V({G'_1}^{\intercal} \odot \ldots \odot {G'_k}^{\intercal} \odot {L'}^{\intercal})\right| \leq D_V, \\ |E_{mid}(G_{k+1})| - |V(G_{k+1})| > CD_{\mathrm{SoS}} \text{ or } |E_{mid}(G'_{k+1})| - |V(G'_{k+1})| > CD_{\mathrm{SoS}}, \\ L, G_{k+1} \circ \ldots \circ G_1 \circ T \circ {G'_1}^{\intercal} \circ \ldots \circ {G'_{k+1}}^{\intercal}, {L'}^{\intercal} \text{ are properly composable,} \\ \text{edge bounds satisfied up to } k}} \lambda_{L \odot G_{k+1} \odot \cdots \odot {G'_{k+1}}^{\intercal} \odot {L'}^{\intercal}} M_{L \circ G_{k+1} \circ \cdots \circ {G'_{k+1}}^{\intercal} \circ {L'}^{T}}$$

This is the source of the truncation error terms with too many edges.

Iteratively applying this procedure gives the result.

Proposition 6.48. *The recursion terminates within $2D_{\mathrm{SoS}}$ steps and $I_{2D_{\mathrm{SoS}}} = 0$.*

Proof. Each additional intersection G_{k+1}, G'_{k+1} is nontrivial and increases $\left|A_{G_{k+1}}\right| + \left|A_{G'_{k+1}}\right|$. Since $\left|A_{G_k}\right|$ is upper bounded by D_{SoS}, the recursion ends within $2D_{\mathrm{SoS}}$ steps. □

□

6.4.2 Overview of handling combinatorial factors

Here we give a preview of the arguments needed to handle combinatorial factors, which will be implemented during the course of the next sections. The charging arguments in Section 6.2.4 show that each individual graph matrix has norm less than 1. Specifically, except for another important combinatorial factor that we will need to return to later, the charging arguments show that:

$$|\lambda_\alpha| \|M_\alpha\| \leq C \cdot \left(\frac{Ck\sqrt{d}}{n} \right)^{|V(\alpha)| - \frac{|U_\alpha| + |V_\alpha|}{2}} .$$

C denotes a universal constant, but one whose value we do not track. Explained in words, we have a "decay factor" of $\frac{Ck\sqrt{d}}{n}$ for each vertex of α, except for vertices in $U_\alpha \cup V_\alpha$. Notably, vertices in $U_\alpha \cap V_\alpha$ have no decay factor at all, so we'll need to be careful about handling vertices in $U_\alpha \cap V_\alpha$.

We would like to prove that the sum of these norms over all non-trivial shapes α is less than $\frac{1}{10}$, for concreteness. To do this, we introduce a function $c(\alpha)$, which is a summable combinatorial tail bound for α. By "summable" we mean that

$$\sum_{\substack{\alpha \in \mathcal{S}: \\ \text{non-trivial}}} \frac{1}{c(\alpha)} \leq \frac{1}{10} .$$

In order to make $c(\alpha)$ summable, the most natural choice of $c(\alpha)$ is to choose any $c(\alpha)$ which is an upper bound on the number of shapes that "match" α. The upper bound $c(\alpha)$ must be carefully chosen to satisfy $\left(\frac{Ck\sqrt{d}}{n}\right)^{|V(\alpha)| - \frac{|U_\alpha| + |V_\alpha|}{2}} \leq \frac{1}{c(\alpha)}$, or equivalently,

$$\left(\frac{n}{Ck\sqrt{d}} \right)^{|V(\alpha)| - \frac{|U_\alpha| + |V_\alpha|}{2}} \geq c(\alpha) .$$

Once we have this, it is clear that $\sum_{\substack{\alpha \in \mathcal{S}: \\ \text{non-trivial}}} |\lambda_\alpha| \|M_\alpha\| \leq \frac{1}{10}$ as desired.

We will now give several upper bounds $c(\alpha)$, starting from a simple choice and working our way up to the final choice of $c(\alpha)$ that is defined formally later in Section 6.4.6. The attempts will explain why we need to introduce various additional complications in the PSD-ness proof in order to control combinatorial factors.

Attempt #1: $c(\alpha) = 2^{|V(\alpha)|^2} C^{|V(\alpha)|}$. This is an upper bound on the number of shapes with $|V(\alpha)|$ vertices, since every such shape can be generated in the following way: for each vertex, pick whether it's in U_α or V_α (4 ways per vertex), then for each pair of vertices, decide whether to include an edge. Therefore, it is clearly summable since we may partition the sum over shapes by the number of vertices in $|V(\alpha)|$, and (for large enough C) we have a decreasing geometric series.

Unfortunately, we cannot prove $\left(\frac{n}{Ck\sqrt{d}}\right)^{|V(\alpha)| - \frac{|U_\alpha| + |V_\alpha|}{2}} \geq c(\alpha)$ with this choice of $c(\alpha)$. We will explain why it doesn't work even with a slightly smarter choice.

Slightly smarter attempt #1: $c(\alpha) = |V(\alpha)|^{2|E(\alpha)|}C^{|V(\alpha)|}$. This is an upper bound on the number of shapes with $|V(\alpha)|$ vertices and $|E(\alpha)|$ edges: each vertex chooses whether to be in U_α or V_α, then each edge chooses its two endpoints. Hence it is summable.

The attempted inequality is $\left(\frac{n}{Ck\sqrt{d}}\right)^{|V(\alpha)|-\frac{|U_\alpha|+|V_\alpha|}{2}} \geq c(\alpha) = |V(\alpha)|^{2|E(\alpha)|}C^{|V(\alpha)|}$. Unfortunately, we do not have decay factors for vertices in $U_\alpha \cap V_\alpha$, and if $|U_\alpha \cap V_\alpha|$ is large while the rest of the shape is small, then $C^{|V(\alpha)|}$ will be too large to control. To fix this, as suggested in Section 2.3, we would like to factor out the dependence on $U_\alpha \cap V_\alpha$.

This choice was successfully used in Planted Clique, [BHK$^+$16]. The difference is that $\frac{n}{Ck\sqrt{d}} = n^\varepsilon$ in the Planted Clique setting, and the size of the shapes is only $O(\log n)$, hence even $C^{|V(\alpha)|}$ can be handled by just one single factor of n^ε. In our setting, we use a smaller decay factor, and also handle larger shapes (ideally up to the "brute force seach" regime $D_{\text{SoS}} = \Omega(\alpha(G))$). We would like the vertex decay factor to be $\frac{n}{Ck\sqrt{d}} = C|U_\alpha \cup V_\alpha|$ (this is not quite the predicted best possible, see the open problems Section 6.5).

Attempt #2: $c(\alpha) = D_{SoS}2^{|U_\alpha \cap V_\alpha|\cdot|W_\alpha|}C^{|V(\alpha)\setminus(U_\alpha\cap V_\alpha)|}|V(\alpha)|^{2|E(\alpha)\setminus(U_\alpha\cap V_\alpha)\times W_\alpha|}$. Proof of upper bound: first, choose how many vertices are in $U_\alpha \cap V_\alpha$, up to D_{SoS}. Then, each vertex in $U_\alpha \cap V_\alpha$ chooses a subset of W_α to be neighbors. Then, each other vertex chooses whether to be U_α or V_α. Then, the remaining edges not incident to $U_\alpha \cap V_\alpha$ choose two endpoints each.

This choice isolates the dependence on $U_\alpha \cap V_\alpha$. Since there is no decay factor for $U_\alpha \cap V_\alpha$, we must allocate any factors that depend on $U_\alpha \cap V_\alpha$ to other vertices. The factor that we must allocate is $2^{|U_\alpha\cap V_\alpha|\cdot|W_\alpha|}$, which naturally may be allocated as $2^{|U_\alpha\cap V_\alpha|}$ per vertex in W_α. Unfortunately, this could be $2^{D_{\text{SoS}}}$ per vertex, which is too much for each vertex in W_α. A factor of $2^{D_{\text{SoS}}}$ per vertex restricts the SoS degree to be relatively small, $D_{\text{SoS}} = O(\log n)$.

To get around this combinatorial factor, we group edges into $U_\alpha \cap V_\alpha$ into *quasi-independent set indicators*. The definition is delayed until Section 6.4.3.

Attempt #3: $c(\alpha) = D_{SoS}(2|W_\alpha| \cdot |U_\alpha \cap V_\alpha|)^{|W_\alpha|}C^{|V(\alpha)\setminus(U_\alpha\cap V_\alpha)|}|V(\alpha)|^{|E(\alpha)|}$ **where α has quasi-independent set indicators.** Proof of upper bound: the same as the previous attempt, except that the choice of edges incident to $U_\alpha \cap V_\alpha$ may be upper bounded by the number of ways to choose quasi-independent set indicators, which is only $(2|W_\alpha| \cdot |U_\alpha \cap V_\alpha|)^{|W_\alpha|}$ (to be shown later).

The dependence on $U_\alpha \cap V_\alpha$ is now under control. Next, we must deal with shapes that have too many edges. The factor $|V(\alpha)|^{|E(\alpha)|}$ may be as large as $|V(\alpha)|^{|V(\alpha)\setminus(U_\alpha\cap V_\alpha)|^2}$. This requires a factor of $|V(\alpha)|^{|V(\alpha)\setminus(U_\alpha\cap V_\alpha)|}$ per vertex, which is too much. We deal with this by *conditioning* the graph. After conditioning, we may assume that all shapes are sparse, i.e. they have $|E(\alpha)| \leq C \cdot |V(\alpha) \setminus (U_\alpha \cap V_\alpha)| + C$.

Now we must pay a factor of approximately $|V(\alpha)|^C$ per vertex. This is pretty good, and it is sufficient for us when $|V(\alpha)| \leq CD_{\text{SoS}}$. However, since $|V(\alpha)|$ could be as large as $D_V = CD_{\text{SoS}} \log n$, this incurs log factors and hinders us from lowering the assumption on the graph degree d below $\log n$ for these larger shapes. We would like to get rid of the log factors and pay only D_{SoS}^C per vertex (this is independent of n e.g. when $D_{\text{SoS}} = 4$). We improve the counting argument (this will also use the connected truncation).

Final attempt #4:

$$c(\alpha) = \begin{cases} \text{same as before} & |V(\alpha)| \leq CD_{SoS} \\ D_{SoS}(2|W_\alpha| \cdot |U_\alpha \cap V_\alpha|)^{|W_\alpha|} C^{|V(\alpha) \setminus (U_\alpha \cap V_\alpha)|} & \\ \quad \cdot (2CD_{SoS})^{2CD_{SoS}} |V(\alpha)|^{2\max(|E(\alpha)| - |V(\alpha)| - CD_{SoS}, 0)} & |V(\alpha)| > CD_{SoS} \end{cases}$$

where α is sparse and has quasi-independent set indicators. Proof of upper bound: same as the prior proof, with the modification that to place the edges of α, we perform the following process.

1. First we place a forest that connects all vertices to $U_\alpha \cup V_\alpha$, which exists due to the connected truncation. A forest can be specified in $3^{2|V(\alpha) \setminus (U_\alpha \cap V_\alpha)|}$ ways: take a walk of length $2|V(\alpha) \setminus (U_\alpha \cap V_\alpha)|$ and specify whether each step goes down the tree to a new vertex, goes up the tree, or starts a new tree in the forest. The remaining edges we call "excess edges".

2. For each of the vertices in $V(\alpha) \setminus (U_\alpha \cap V_\alpha)$, we specify whether or not it is adjacent to one of the first CD_{SoS} excess edges of the shape (2 ways per vertex).

3. For each of the first CD_{SoS} excess edges of the shape, we specify its endpoints among the just-selected vertex set, in at most ways $(2CD_{\text{SoS}})^2$ per edge.

4. For the remaining edges, we naïvely choose its endpoints in $|V(\alpha)|^2$ ways.

Using this choice of $c(\alpha)$, we only need to pay D_{SoS}^C per vertex for the first CD_{SoS} edges, as desired. The only remaining case is when the shape has many vertices and many excess edges, in which case we will start to pay $|V(\alpha)|^2$ per edge, which is too much. For these shapes, we improve the upper bound on $\|M_\alpha\|$ by giving a tighter bound on the *Frobenius norm* of M_α. This allows us to show that these large shapes have small norm.

Other combinatorial factors We have neglected a combinatorial factor in the norm bound of the form approximately $D_V^{|V(\alpha)| - \frac{|U_\alpha| + |V_\alpha|}{2}}$. Since this equals $CD_{\text{SoS}} \log n$ per vertex, this factor requires one log factor per vertex to handle. There are also several other combinatorial factors that crop up, but they are bounded by factors of D_{SoS} per vertex. Hence in total, we require a decay factor of $D_{\text{SoS}}^C \log n$, which is the source of the assumption in the main theorem that

$$k \leq \frac{n}{D_{\text{SoS}}^C \cdot \log n \cdot d^{1/2}}.$$

6.4.3 Factor out Π and edges incident to $U_\alpha \cap V_\alpha$

The moment matrix Λ has a null space that we need to factor out. This is because Λ needs to satisfy the independent set constraints ($\widetilde{\mathbb{E}}[x^S] = 0$ if S has an edge). For independent set it's easy to factor out the null space, whereas handling the null space was the main challenge in Chapter 4.

We need to augment our graph matrices to include missing edge indicators. There are two reasons. First, the dominant term in the decomposition is a projection matrix with these indicators, rather than the identity matrix. Second, we need to factor out dependence on $U_\tau \cap V_\tau$. These vertices do not essentially participate in the graph matrix, since the matrix is block diagonal with a block for each assignment to $U_\alpha \cap V_\alpha$. (When proving things with graph matrices, it is smart to start by assuming $U_\alpha \cap V_\alpha = \emptyset$, then handle the case $U_\alpha \cap V_\alpha \neq \emptyset$).

For the first issue, define the matrix Π.

Definition 6.49. *Let $\pi \in \mathbb{R}^{\binom{n}{\le D_{\mathrm{SoS}}} \times \binom{n}{\le D_{\mathrm{SoS}}}}$ be the projector to the independent set constraints. π is a diagonal matrix with entries*

$$\pi[S,S] = \mathbb{1}[S \textit{ is an independent set in } G]$$

Definition 6.50. *Let $\Pi \in \mathbb{R}^{\binom{n}{\le D_{\mathrm{SoS}}} \times \binom{n}{\le D_{\mathrm{SoS}}}}$ be a rescaling of π by*

$$\Pi[S,S] = \left(\frac{1}{1-p}\right)^{\binom{|S|}{2}} \mathbb{1}[S \textit{ is an independent set in } G]$$

The rescaling satisfies $\mathbb{E}_{G \sim G_{n,p}}[\Pi[S,S]] = 1$. Recall that in Proposition 6.10, we had the function $1 + \sqrt{\frac{p}{1-p}}\chi_{\{e\}} = \frac{1}{1-p}\mathbb{1}_{e \notin E(G)}$ which is $\frac{1}{1-p}$ times the indicator function for e being absent from the input graph G. Since the coefficients λ_R come with $\sqrt{\frac{p}{1-p}}$ for each edge, we can group the edges inside A_R, B_R into missing edge indicators for all edges inside A_R, B_R – that is, independent set indicators on A_R, B_R which form the matrix Π. For all the sums of ribbons we consider, after grouping:

$$\sum_R \lambda_R M_R = \Pi^{1/2} \left(\sum_{\substack{R: \\ E(A_R)=E(B_R)=\emptyset}} \left(\frac{1}{1-p}\right)^{\binom{|A_R|}{2}/2 + \binom{|B_R|}{2}/2 - \binom{|A_R \cap B_R|}{2}} \lambda_R M_R \right) \Pi^{1/2}.$$

For the second issue, consider the function $1 + \sqrt{\frac{p}{1-p}}\chi_{\{e\}} = \frac{1}{1-p}\mathbb{1}_{e \notin E(G)}$ again. The magnitude of this function is clearly bounded by $\frac{1}{1-p}$. However, if we try to bound the magnitude of this function term by term, we instead get a bound of 2 because 1 always has magnitude 1 and if $e \in E(G)$ then $\sqrt{\frac{p}{1-p}}\chi_{\{e\}} = 1$, so the best bound we can give on

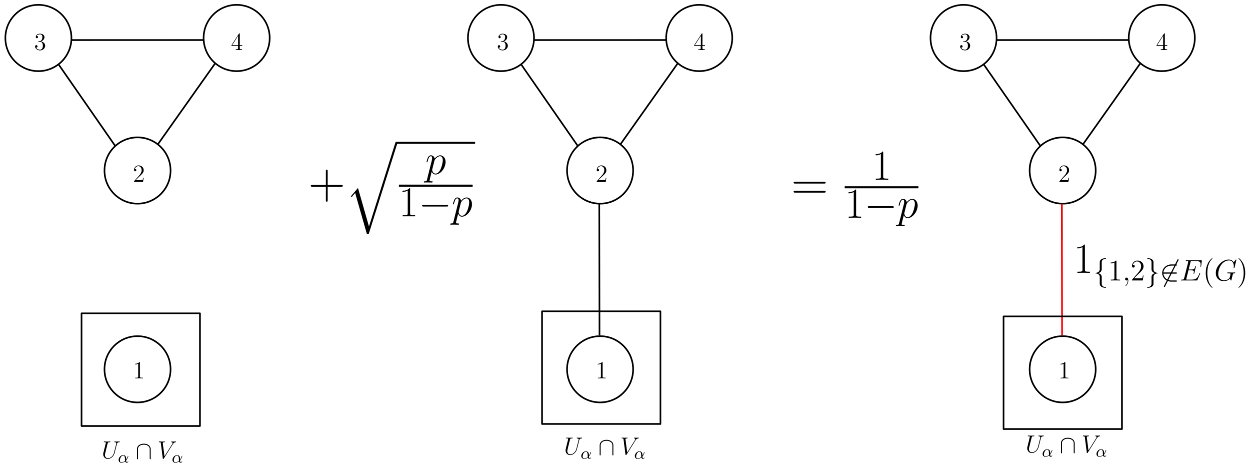


Figure 6.3: The two ribbons on the left side can be added together to get a multiple of the indicator function for the edge $\{1,2\}$, which is the red edge on the right. However, the leftmost ribbon does not appear due to the connected truncation.

the magnitude of $\sqrt{\frac{p}{1-p}}\chi_{\{e\}}$ is also 1.

This can become a problem if $U_\tau \cap V_\tau$ is large, in which case there are many possible subsets of edges incident to $U_\tau \cap V_\tau$ even if the rest of the shape is small. If we try to bound things term by term, we will get a factor of $2^{|U_\tau \cap V_\tau|}$ which may be too large. To handle this, our strategy will be to group terms into missing edge indicators, which gives a factor of $\left(\frac{1}{1-p}\right)^{|U_\tau \cap V_\tau|}$ per vertex instead. This almost works, however due to the connected truncation there are some edge cases of Proposition 6.10 where we do not have the term where T is empty. For example, see Fig. 6.3. We handle this using the following lemma.

Lemma 6.51. *For any set of potential edges E,*

$$\sum_{E'\subseteq E:E'\neq\emptyset}\left(\sqrt{\frac{p}{1-p}}\right)^{|E'|}\chi_{E'} = \sum_{e\in E}\left(\sqrt{\frac{p}{1-p}}\right)\chi_{\{e\}}\sum_{E'\subseteq E\setminus\{e\}}\frac{1}{|E|\binom{|E|-1}{|E'|}}\left(\frac{1}{1-p}\right)^{|E'|}\mathbb{1}_{\forall e'\in E',e'\notin E(G)}$$

Proof. Observe that if we give an ordering to the edges of E then we can write

$$\sum_{E'\subseteq E:E'\neq\emptyset}\left(\sqrt{\frac{p}{1-p}}\right)^{|E'|}\chi_{E'} = \sum_{e\in E}\left(\sqrt{\frac{p}{1-p}}\right)\chi_{\{e\}}\left(\prod_{e'\in E:e'>e}\left(\frac{1}{1-p}\right)\mathbb{1}_{e'\notin E(G)}\right)$$

Taking the average over all orderings of the edges of E gives the result. To see this, observe that if we take a random ordering, the probability of having a term $\left(\sqrt{\frac{p}{1-p}}\right)\chi_{\{e\}}\cdot\left(\frac{1}{1-p}\right)^{|E'|}\mathbb{1}_{\forall e'\in E',e'\notin E(G)}$ is $\frac{1}{|E|\binom{|E|-1}{|E'|}}$ as there must be exactly $|E'|$ elements after e (so e

must be in the correct position) and these elements must be E'. □

Remark 6.52. *We take the average over all orderings of the edges E so that our expression will be symmetric with respect to permuting the indices in $U_\tau \cap V_\tau$.*

Based on this lemma, we make the following definition.

Definition 6.53 (Quasi-indicator function)**.** *Given a set of edges $E \subseteq \binom{[n]}{2}$, we define the quasi-missing edge indicator function q_E to be*

$$q_E = (1-p)^{|E|+1} \sum_{E' \subseteq E} \frac{1}{(|E|+1)\binom{|E|}{|E'|}} \left(\frac{1}{1-p}\right)^{|E'|} \mathbb{1}_{\forall e \in E', e \notin E(G)}$$

Proposition 6.54. $\sum_{E' \subseteq E : E' \neq \emptyset} \left(\sqrt{\frac{p}{1-p}}\right)^{|E'|} \chi_{E'} = \left(\frac{1}{1-p}\right)^{|E|} \sum_{e \in E} \left(\sqrt{\frac{p}{1-p}}\right) \chi_{\{e\}} q_{E \setminus \{e\}}$

Remark 6.55. *q_E is a linear combination of terms where for each edge $e \in E$, either there is a missing edge indicator for e or e is not mentioned at all. Moreover, we add the factor of $(1-p)^{|E|+1}$ to q_E so that the coefficients in this linear combination are non-negative and have sum at most 1. This is the only fact that we will use about q_E when we analyze the norms of the resulting graph matrices.*

Definition 6.56 (Augmented ribbon)**.** *An augmented ribbon is a vertex set $V(R) \subseteq [n]$ with two subsets $A_R, B_R \subseteq V(R)$, as well as a multiset of edge-functions. In our augmentation, each edge-function is either a (single edge) Fourier character, a (single edge) missing edge indicator, or a (subset of edges) quasi-missing edge indicator.*

Definition 6.57 (Matrix for an augmented ribbon)**.** *The matrix M_R has rows and columns indexed by all subsets of $[n]$, with entries:*

$$M_R[I,J] = \begin{cases} \prod\limits_{\textit{edge-functions } f \textit{ on } S \subseteq V(R)} f(G|_S) & I = A_R, J = B_R \\ 0 & \textit{Otherwise} \end{cases}$$

Two augmented ribbons are isomorphic if they are in the same S_n-orbit (i.e. they are equal after renumbering the vertices), and we define an augmented shape for each orbit. Equivalently, augmented shapes are equivalence classes of augmented ribbons under *type-preserving* multi-hypergraph isomorphism, where the type of a hyperedge is the associated function. As before, we can specify the shape by a representative graph with edge-functions. The graph matrix M_α for an augmented shape α is still defined as the sum of M_R over injective embeddings of α into $[n]$.

Definition 6.58 (Permissible)**.** *We say that an augmented ribbon R is permissible if the following conditions are satisfied:*

1. *All vertices $v \in V(R)$ are reachable from $A_R \cup B_R$ using the Fourier character edges.*

2. *There is a missing edge indicator for all edges within A_R and a missing edge indicator for all edges within B_R.*

3. *For any vertex $v \in V(R) \setminus (A_R \cap B_R)$ which is reachable from $(A_R \cup B_R) \setminus (A_R \cap B_R)$ without passing through $(A_R \cap B_R)$, there is a missing edge indicator for all edges between $(A_R \cap B_R)$ and v.*

4. *For each connected component C of $R \setminus (A_R \cap B_R)$ which is disconnected from $A_R \cup B_R$, there is precisely one pair of vertices $u \in A_R \cap B_R$ and $w \in C$ such that $(u, w) \in E(R)$ and R contains the quasi-missing edge indicator $q_{((A_R \cap B_R) \times C) \setminus \{u,w\}}$ for all other edges between $A_R \cap B_R$ and C.*

5. *There are no other (quasi-)missing edge indicators, and the Fourier character edges are disjoint from the (quasi-)missing edge indicators.*

A shape is permissible if any (equivalently, all) of its ribbons are permissible.

An example is shown in Fig. 6.4. To explain the conditions, the first condition is the connected truncation. The second condition follows because we factored out independent set indicators. The third condition observes that if we have a vertex $v \in W_\alpha$ that is connected to $(U_\alpha \cup V_\alpha) \setminus (U_\alpha \cap V_\alpha)$, then we can factor out a missing edge indicator for all edges that go from v to $U_\alpha \cap V_\alpha$ because such edges do not affect the connectivity properties of α. The fourth condition picks out components of W_α that do affect the connectivity and handles them via the definition of quasi-missing edge indicators as explained above.

Remark 6.59. *For the remainder of Section 6.4, ribbon means "augmented ribbon" and shape means "augmented shape". $E(\alpha)$ refers to the multiset of Fourier characters edges in α. We will write "permissible $R \in \mathcal{S}$" to mean a permissible ribbon such that the corresponding non-augmented ribbon that includes all edges involved in any edge function is in $\mathcal{S}$ (and similarly for other sets of ribbons/shapes).*

Lemma 6.60. *If α is permissible, then $\pi M_\alpha = M_\alpha \pi = M_\alpha$.*

Proof. α has missing edge indicators for all edges inside U_α and V_α, hence M_α is zero on any rows or columns that are not independent sets. □

We count the extra factors of $1/(1-p)$ with the following definition.

Definition 6.61. *For a permissible shape or ribbon R, let:*

$$m(R) = \left(\frac{1}{1-p}\right)^{\binom{|A_R|}{2}/2 + \binom{|B_R|}{2}/2 - \binom{|A_R \cap B_R|}{2} + |V(R) \setminus (A_R \cup B_R)| \cdot |A_R \cap B_R|}.$$

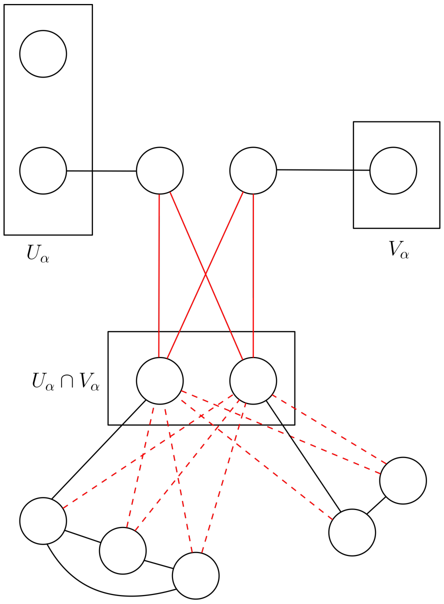


Figure 6.4: Example of a permissible shape. The solid red edges are independent set indicators. The dashed red edges represent two quasi-missing edge indicators between $U_\alpha \cap V_\alpha$ and the two "floating" connected components of $V(\alpha) \setminus (U_\alpha \cap V_\alpha)$. Not shown are missing edge indicators between all pairs of vertices in U_α and all pairs of vertices in V_α.

Lemma 6.62.

$$\sum_{R \in \mathcal{S}} \lambda_R M_R =$$

$$\Pi^{1/2} \left(\sum_{\substack{\textit{permissible} \\ L \in \mathcal{L}}} m(L) \lambda_L M_L \right) \cdot$$

$$\left(\sum_{j=0}^{2D_{\mathsf{SoS}}} (-1)^j \sum_{\substack{\textit{permissible} \\ G_j, \ldots, G_1, T, G'_1, \ldots, G'_j}} m(T) \left(\prod_{i=1}^{j} m(G_i) m(G'_i) \right) \lambda_{G_j \odot \ldots \odot G_1 \odot T \odot {G'_1}^{\mathsf{T}} \odot \ldots \odot {G'_j}^{\mathsf{T}}} M_{G_j \circ \ldots \circ G_1 \circ T \circ {G'_1}^{\mathsf{T}} \circ \ldots \circ {G'_j}^{\mathsf{T}}} \right) \cdot$$

$$\left(\sum_{\substack{\textit{permissible} \\ L \in \mathcal{L}}} m(L) \lambda_L M_L \right)^{\mathsf{T}} \Pi^{1/2}$$

$$+ \textit{ truncation error}$$

Proof. The factoring process is completely independent of edges inside $A_T, B_T, A_{G_i}, B_{G_i}$

or edges incident to $A_R \cap B_R$ for any of these shapes, as noted in Lemma 6.47. All subsets of edges inside A_R or B_R are present, but not all subsets of edges incident to $A_R \cap B_R$ are present; because of the connected truncation, we do not have the empty set of edges between $A_R \cap B_R$ and components of $R \setminus (A_R \cap B_R)$ which are disconnected from $A_R \cup B_R$. Now group these Fourier characters into missing edge indicators and quasi-missing edge indicators as explained above. □

We can show the factor $m(R)$ is a negligible constant:

Lemma 6.63. *For any ribbon R with degree at most D_{SoS},*

$$m(R) \leq \left(\frac{1}{(1-p)^{2D_{\mathrm{SoS}}}}\right)^{|V(R)\setminus(A_R \cap V_R)|}.$$

Proof. The claim follows from the following two inequalities:

$$\binom{|A_R|}{2}/2 + \binom{|B_R|}{2}/2 - \binom{|A_R \cap B_R|}{2} \leq D_{\mathrm{SoS}}(|A_R \setminus (A_R \cap B_R)| + |B_R \setminus (A_R \cap B_R)|)$$

$$|A_R \cap B_R||V(R) \setminus (A_R \cup B_R)| \leq D_{\mathrm{SoS}}|V(R) \setminus (A_R \cap B_R)|.$$

□

This will be handled later by the vertex decay.

6.4.4 Conditioning I: reduction to sparse shapes

To improve norm bounds for dense shapes, we can replace them by sparse ones. The replacement of dense ribbons is accomplished by the following lemmas.

Lemma 6.64. $\mathbb{1}_{e \in E(G)}\chi_e = -\sqrt{p(1-p)} + (1-p)\chi_e$

Proof. Follows by explicit computation, as in Proposition 6.10. □

Lemma 6.65. *Given a set of edges E, if we know that not all of the edges of E are in $E(G)$ then*

$$\chi_E = -\sum_{E' \subseteq E : E' \neq E} \left(-\sqrt{\frac{p}{1-p}}\right)^{|E|-|E'|} \chi_{E'}$$

Proof. Since not all of the edges of E are in $E(G)$, $(1 - \mathbb{1}_{E \subseteq E(G)})\chi_E = \chi_E$. Now observe that

$$\mathbb{1}_{E \subseteq E(G)}\chi_E = \prod_{e \in E} (\mathbb{1}_{e \in E(G)}\chi_e) = \prod_{e \in E} (-\sqrt{p(1-p)} + (1-p)\chi_e)$$

Thus,

$$\chi_E = (1 - \mathbb{1}_{E \subseteq E(G)})\chi_E = (1-(1-p)^{|E|})\chi_E - \sum_{E' \subseteq E : E' \neq E} (-\sqrt{p(1-p)})^{|E|-|E'|}(1-p)^{|E'|}\chi_{E'}$$

Solving for χ_E gives

$$\chi_E = - \sum_{E' \subseteq E : E' \neq E} \left(- \sqrt{\frac{p}{1-p}} \right)^{|E|-|E'|} \chi_{E'}$$

□

Corollary 6.66. *Given a set of edges E, if we know that at most $k < |E|$ of the edges of E are in $E(G)$, then*

$$\chi_E = (-1)^{|E|-k} \sum_{E' \subseteq E : |E'| \leq k} \binom{|E| - 1 - |E'|}{|E| - 1 - k} \left(- \sqrt{\frac{p}{1-p}} \right)^{|E|-|E'|} \chi_{E'}.$$

Proof. Apply Lemma 6.65 repeatedly. □

Corollary 6.67. *Given a set of edges E, if we know that at most $k < |E|$ of the edges of E are in $E(G)$, then*

$$\chi_E = \sum_{E' \subseteq E : |E'| \leq k} c_{E'} \chi_{E'}$$

where $|c_{E'}| \leq (2|E|\sqrt{p})^{|E|-|E'|}$.

We can do the replacement if the graph of the shape does not occur in our random sample of $G \sim G_{n,p}$. We formalize a bound on the density of subgraphs of $G_{n,p}$. In expectation the number of occurrences of a small subgraph H is essentially $n^{|V(H)|} p^{|E(H)|}$ and hence H is unlikely to appear in a random graph sample if it has too many edges (specifically, morally all subgraphs are sparse, $|E(H)| \leq |V(H)| \log_{1/p}(n)$). To translate from expectation to concentration, we use a simple first moment calculation to bound the densest-k-subgraph in $G_{n,p}$.

Proposition 6.68 (Sparsity of small subgraphs of $G_{n,p}$)**.** *For $G \sim G_{n,p}$, $p \leq \frac{1}{2}$, constant $\eta > 0$, with probability at least $1 - O(1/n^{\eta})$, every subgraph S of G such that $|V(S)| \leq \sqrt[3]{\frac{n}{d}}$ satisfies:*

$$|E(S)| \leq 3|V(S)| \log_{1/p}(n) + 3\eta \log_{1/p}(n).$$

Proof. Let $e^*(v) := 3v\log_{1/p}(n) + 3\eta\log_{1/p}(n)$.

$$\begin{aligned}
\Pr[\exists\text{too dense subgraph}] &\le \sum_{v=2}^{n}\sum_{e=e^*(v)}^{\binom{v}{2}} \mathbb{E}[\text{number of subgraphs with } v \text{ vertices, } e \text{ edges}] \\
&\le \sum_{v=2}^{n}\sum_{e=e^*(v)}^{\binom{v}{2}} \binom{n}{v} v^{2e} p^e \\
&\le \sum_{v=2}^{n}\sum_{e=e^*(v)}^{\binom{v}{2}} \frac{n^v}{v!}(p^{1/3})^e \\
&= \sum_{v=2}^{n}\sum_{e=e^*(v)}^{\binom{v}{2}} \frac{n^v p^{e^*(v)/3}}{v!}(p^{1/3})^{e-e^*(v)} \\
&= \sum_{v=2}^{n}\sum_{e=e^*(v)}^{\binom{v}{2}} \frac{1}{n^\eta \cdot v!}(p^{1/3})^{e-e^*(v)} \\
&\le O(1/n^\eta).
\end{aligned}$$

□

Definition 6.69 (Sparse). *A graph G is C'-sparse if $|E(G)| \le C'|V(G)|$. A shape α is C'-sparse if the underlying graph (not multigraph) is. We will generally say that a graph/shape is sparse if it is C'-sparse for some C'.*

Corollary 6.70. *For $d \le n^{0.5}$, w.h.p. every subgraph of $G \sim G_{n,\frac{d}{n}}$ of size at most $n^{0.16}$ is 7-sparse.*

Definition 6.71 (Forbidden subgraph). *We call a graph H forbidden if it does not occur as a subgraph of G.*

The assumptions in Theorem 6.3 ensure that every graph of size at most D_V that is not sparse is forbidden per Corollary 6.70. This is the only class of forbidden graphs that we will need in this work.

Remark 6.72. *For $d = n^{1-\varepsilon}$, subgraphs up to size $n^{\Omega(\varepsilon)}$ will be $O(1/\varepsilon)$-sparse. Using this bound and our techniques, Theorem 6.3 can be extended to show a $n^{\Omega(\varepsilon)}$ SoS-degree lower bound for $d = n^{1-\varepsilon}$.*

Remark 6.73. *For $d \ll n^{\varepsilon}$, subgraphs of $G_{n,p}$ will actually be significantly sparser. For example, it is well-known that $o(\log n)$-radius neighborhoods in $G_{n,d/n}$ for constant d are trees with at most one extra cycle.*

Definition 6.74 (Subshape/subribbon, supershape/superribbon). *We call shape β a subshape of shape α if $V(\alpha) = V(\beta)$, $U_\alpha = U_\beta, V_\alpha = V_\beta$, and $E(\beta) \subseteq E(\alpha)$. Furthermore, the* *(quasi-)missing edge indicators of β and α must be equal. Supershapes, subribbons, and superribbons are defined similarly. We write $\alpha \subseteq \beta$ if α is a subshape of β.*

Definition 6.75. *For a sparse subribbon R of a forbidden ribbon U, let $c(R, U)$ be the coefficient on R after applying conditioning Corollary 6.66 to the non-reserved edges of U.*

Definition 6.76. *Let λ'_R incorporate the constant factors into λ_R,*[6]

$$\begin{aligned} \lambda'_R &:= m(R)\left(\sum_{U \supseteq R} c(R,U)\lambda_U\right)\lambda_R, \\ \lambda'_{R_1 \odot \cdots \odot R_k} &:= \prod_{i=1}^{k} \lambda'_{R_i}. \end{aligned}$$

Lemma 6.77.

$$\begin{aligned} &\sum_{R \in \mathcal{S}} \lambda_R M_R = \\ &\Pi^{1/2}\left(\sum_{\substack{\textit{sparse}\\ \textit{permissible}\\ L \in \mathcal{L}}} \lambda'_L M_L\right) \cdot \\ &\left(\sum_{j=0}^{2D_{\mathrm{SoS}}} (-1)^j \sum_{\substack{\textit{sparse}\\ \textit{permissible}\\ G_j, \ldots, G_1, T, G'_1, \ldots, G'_j}} \lambda'_{G_j \odot \ldots \odot G_1 \odot T \odot G_1'^{\mathsf{T}} \odot \ldots \odot G_j'^{\mathsf{T}}} M_{G_j \circ \ldots \circ G_1 \circ T \circ G_1'^{\mathsf{T}} \circ \ldots \circ G_j'^{\mathsf{T}}}\right) \cdot \\ &\left(\sum_{\substack{\textit{sparse}\\ \textit{permissible}\\ L \in \mathcal{L}}} \lambda'_L M_L\right)^{\mathsf{T}} \Pi^{1/2} \\ &+ \textit{truncation error} \end{aligned}$$

The proof follows mostly by definition of λ'_R but there is an important detail. When we condition a left L/middle T/intersection G ribbon, the result may no longer be a left/middle/intersection ribbon. For example, the MVS might change drastically. It turns out that we can avoid this by "reserving" $O(|V(R)|)$ edges in the ribbon and only allowing the removal of edges outside of the reserved set. The reserved edges guarantee that

[6] When R is a left L /middle T/intersecting G ribbon, the sum over U should be restricted to U with the same type, and which actually appear in the decomposition. The stated sum is an upper bound.

the subribbon will continue to be a left/middle/intersection ribbon, at the cost of increasing the sparsity. As shown in [JPR$^+$21, Appendix B], $O(|V(R)|)$ edges suffice for each of the different types of ribbons.

We can show that the conditioning negligibly affects the coefficients of the ribbons.

Lemma 6.78. *Under the conditions of Theorem 6.3, for any ribbon $R \in \mathcal{S}$,*

$$\sum_{U \supset R} \lambda_U c(R, U) = o(1)\lambda_R.$$

Proof. Let U be a superribbon with k more edges than R. We have

$$\lambda_U = \left(\sqrt{\frac{p}{1-p}}\right)^k \lambda_U \leq (2\sqrt{p})^k \lambda_R. \tag{6.1}$$

Using the bound from Corollary 6.67,

$$|c(R, U)| \leq (2|E(U)|\sqrt{p})^k \leq (2|V(R)|^2\sqrt{p})^k. \tag{6.2}$$

The number of ribbons U that are superribbons of R with k extra edges is at most

$$|V(R)|^{2k} \tag{6.3}$$

because (see [JPR$^+$21, Appendix B]) due to the connectivity of the reserved set, the vertex set of U and R is the same. Combining Eq. (6.1), Eq. (6.2), Eq. (6.3), the change in coefficient is at most

$$\sum_{k=1}^{\infty} \lambda_R (4|V(R)|^4 p)^k \leq \sum_{k=1}^{\infty} \lambda_R \left(\frac{4D_V^4 d}{n}\right)^k \leq o(1)\lambda_R.$$

The last inequality uses $d \leq n^{0.5}$ and $D_V \leq \widetilde{O}(D_{\text{SoS}}) \leq \widetilde{O}(n^{0.1})$. □

6.4.5 *Shifting to shapes*

For a proper middle shape τ and left shapes $\gamma_j, \ldots, \gamma_1, \gamma_1', \ldots, \gamma_j'$, recall the notation $\mathcal{P}^{mid}_{\gamma_j, \ldots, \gamma_1, \tau, \gamma_1', \ldots, \gamma_j'}$ for the set of middle intersection patterns (Definition 6.20 and Remark 6.22). We also let $\mathcal{P}^{mid}_{\tau}$ contain a single term, the non-intersecting singleton partition.

We now analyze

$$\sum_{\substack{\text{sparse} \\ \text{permissible} \\ G_j, \ldots, G_1, T, G_1', \ldots, G_j'}} \lambda'_{G_j \odot \ldots \odot G_1 \odot T \odot G_1'^{\intercal} \odot \ldots \odot G_j'^{\intercal}} M_{G_j \circ \ldots \circ G_1 \circ T \circ G_1'^{\intercal} \circ \ldots \circ G_j'^{\intercal}}$$

We first partition this sum based on the shapes $\gamma_j, \ldots, \gamma_1, \tau, \gamma_1', \ldots, \gamma_j'$ of $G_j, \ldots, G_1, T, G_1', \ldots, G_j'$. We then partition this sum further based on the intersection pattern $P \in \mathcal{P}^{mid}_{\gamma_j,\ldots,\gamma_1,\tau,\gamma_1',\ldots,\gamma_j'}$.

Definition 6.79. *Define $N_P(\tau_P)$ to be the number of ways to choose ribbons $G_j, \ldots, G_1, T, G_1', \ldots, G_j'$ of shapes $\gamma_j, \ldots, \gamma_1, \tau, \gamma_1', \ldots, \gamma_j'$ so that they have intersection pattern P and $G_j \circ \ldots \circ G_1 \circ T \circ G_1'^{\intercal} \circ \ldots \circ G_j'^{\intercal} = T_P$ for a given ribbon T_P of shape τ_P.*

By symmetry, this is independent of the choice of T_P.

Recall that an intersection pattern specifies both intersections and how the shapes should be glued together via bijections between the V of each shape and the U of the following shape (Definition 2.37).

Definition 6.80. *We say that two intersection patterns are equivalent if there is an element of*

$$Aut(\gamma_j) \times \ldots \times Aut(\gamma_1) \times Aut(\tau) \times Aut(\gamma_1'^{\intercal}) \times \ldots \times Aut(\gamma_j'^{\intercal})$$

which maps one intersection pattern to the other (where the gluing maps are permuted accordingly).

This gives us the following equation:

$$\begin{aligned}&\sum_{\substack{\text{sparse}\\\text{permissible}\\G_j,\ldots,G_1,T,G_1',\ldots,G_j'}} \lambda'_{G_j \odot \ldots \odot G_1 \odot T \odot G_1'^{\intercal} \odot \ldots \odot G_j'^{\intercal}} M_{G_j \circ \ldots \circ G_1 \circ T \circ G_1'^{\intercal} \circ \ldots \circ G_j'^{\intercal}}\\
= &\sum_{\substack{\text{sparse}\\\text{permissible}\\\gamma_j,\ldots,\gamma_1,\tau,\gamma_1',\ldots,\gamma_j'}} \sum_{\substack{\text{nonequivalent}\\P \in \mathcal{P}^{mid}_{\gamma_j,\ldots,\gamma_1,\tau,\gamma_1',\ldots,\gamma_j'}}} N_P(\tau_P) \lambda'_{\gamma_j \circ \cdots \circ \gamma_1 \circ \tau \circ \gamma_1'^{\intercal} \circ \cdots \circ \gamma_j'^{\intercal}} \frac{M_{\tau_P}}{|\operatorname{Aut}(\tau_P)|}\end{aligned}$$

Grouping into shapes, here is the full decomposition:

Lemma 6.81 (Decomposition in terms of shapes)**.**

$$\sum_{R\in\mathcal{S}}\lambda_R M_R =$$

$$\Pi^{1/2}\left(\sum_{\substack{\textit{sparse}\\\textit{permissible}\\\sigma\in\mathcal{L}}}\lambda'_{\sigma}\frac{M_\sigma}{|\operatorname{Aut}(\sigma)|}\right)\cdot$$

$$\left(\sum_{j=0}^{2D_{\text{SoS}}}(-1)^j\sum_{\substack{\textit{sparse}\\\textit{permissible}\\\gamma_j,\ldots,\gamma_1,\tau,\gamma'_1,\ldots,\gamma'_j}}\sum_{\substack{\textit{nonequivalent}\\P\in\mathcal{P}^{mid}_{\gamma_j,\ldots,\gamma_1,\tau,\gamma'_1,\ldots,\gamma'_j}}}N_P(\tau_P)\lambda'_{\gamma_j\circ\cdots\circ\gamma_1\circ\tau\circ\gamma_1'^{\intercal}\circ\cdots\circ\gamma_j'^{\intercal}}\frac{M_{\tau_P}}{|\operatorname{Aut}(\tau_P)|}\right)\cdot$$

$$\left(\sum_{\substack{\textit{sparse}\\\textit{permissible}\\\sigma\in\mathcal{L}}}\lambda'_{\sigma}\frac{M_\sigma}{|\operatorname{Aut}(\sigma)|}\right)^{\intercal}\Pi^{1/2}$$

$$+\,\Pi^{1/2}\textit{truncation error}_{\textit{too many vertices}}\Pi^{1/2}+\Pi^{1/2}\textit{truncation error}_{\textit{too many edges in one part}}\Pi^{1/2}$$

where

1. $$\textit{truncation error}_{\textit{too many vertices}} = -\sum_{\substack{\textit{sparse, permissible}\\\sigma,\tau,\sigma':\\|V(\sigma\circ\tau\circ\sigma'^{\intercal})|>D_V}}\lambda'_{\sigma\circ\tau\circ\sigma'^{\intercal}}\frac{M_{\sigma\circ\tau\circ\sigma'^{\intercal}}}{|\operatorname{Aut}(\sigma\circ\tau\circ\sigma'^{\intercal})|}$$

$$+\sum_{j=1}^{2D_{\text{SoS}}}(-1)^{j+1}\sum_{\substack{\textit{sparse, permissible}\\\sigma,\gamma_j,\ldots,\gamma_1,\tau,\gamma'_1,\ldots,\gamma'_j,\sigma':\\|V(\sigma\circ\gamma_j\circ\ldots\circ\gamma_1)|>D_V\ \textit{or}\\|V(\gamma_1'^{\intercal}\circ\ldots\circ\gamma_j'^{\intercal}\circ\sigma'^{\intercal})|>D_V}}\sum_{\substack{\textit{nonequiv.}\\P\in\mathcal{P}^{mid}_{\gamma_j,\ldots,\gamma'_j}}}N_P(\tau_P)\lambda'_{\sigma\circ\gamma_j\circ\ldots\circ\gamma_j'^{\intercal}\circ\sigma'^{\intercal}}\frac{M_\sigma}{|\operatorname{Aut}(\sigma)|}\frac{M_{\tau_P}}{|\operatorname{Aut}(\tau_P)|}\frac{M^{\intercal}_{\sigma'}}{|\operatorname{Aut}(\sigma')|}$$

$$\textit{truncation error}_{\textit{too many edges in one part}} = \sum_{\substack{\textit{sparse, permissible} \\ \sigma,\tau,\sigma': \\ |V(\sigma\circ\tau\circ\sigma'^{\intercal})|\leq D_V, \\ |E_{mid}(\tau)|-|V(\tau)|>CD_{\text{SoS}}}} \lambda'_{\sigma\circ\tau\circ\sigma'^{\intercal}} \frac{M_{\sigma\circ\tau\circ\sigma'^{T}}}{|\operatorname{Aut}(\sigma\circ\tau\circ\sigma'^{\intercal})|}$$

$$+ \sum_{j=1}^{2D_{\text{SoS}}} (-1)^j \sum_{\substack{\textit{sparse, permissible} \\ \sigma,\gamma_j,\ldots,\gamma_1,\tau,\gamma'_1,\ldots,\gamma'_j,\sigma': \\ |V(\sigma\circ\gamma_j\circ\ldots\circ\gamma_1)|\leq D_V \textit{ and } |V(\gamma_1'^{\intercal}\circ\cdots\circ\gamma_j'^{\intercal}\circ\sigma'^{\intercal})|\leq D_V \\ |E_{mid}(\gamma_j)|-|V(\gamma_j)|>CD_{\text{SoS}} \textit{ or } |E_{mid}(\gamma'_j)|-|V(\gamma'_j)|>CD_{\text{SoS}}}} \sum_{\substack{\textit{nonequiv.} \\ P\in\mathcal{P}^{mid}_{\gamma_j,\cdots,\gamma'_j}}} \lambda'_{\sigma\circ\gamma_j\circ\cdots\circ\gamma_j'^{\intercal}\circ\sigma'^{\intercal}} \frac{M_{\sigma\circ\tau_P\circ\sigma'^{\intercal}}}{|\operatorname{Aut}(\sigma\circ\tau_P\circ\sigma'^{\intercal})|}$$

2. *in all of these sums, the shapes $\sigma, \gamma_j, \ldots, \gamma_1, \tau, \gamma'_1, \ldots, \gamma'_j, \sigma'$ satisfy the following conditions:*

   (a) *$\tau \in \mathcal{M}$, $\sigma, \sigma' \in \mathcal{L}$, and each $\gamma_i, \gamma'_i \in \mathcal{L}$.*

   (b) *$\sigma, \gamma_j, \ldots, \gamma_1, \tau, \gamma_1'^{\intercal}, \ldots, \gamma_j'^{\intercal}, \sigma'^{\intercal}$ are composable.*

   (c) *$|V(\tau)| \leq D_V$, $|V(\gamma_j \circ \ldots \circ \gamma_1)| \leq D_V$, and $|V(\gamma_1'^{\intercal} \circ \ldots \circ \gamma_j'^{\intercal})| \leq D_V$*

   (d) *Except when noted otherwise (which only happens for truncation error$_{\textit{too many edges in one part}}$), all of the shapes $\gamma_j, \ldots, \gamma_1, \tau, \gamma'_1, \ldots, \gamma'_j$ (but not necessarily σ, σ') satisfy $|E_{mid}(\alpha)| - |V(\alpha)| \leq CD_{\text{SoS}}$.*

To analyze these expressions, we need upper bounds on the number of possible intersection patterns, and on $N_P(\tau_P)$. To gain slightly more control we furthermore partition the intersection patterns based on how many intersections occur. The proofs of these lemmas are found in [JPR+21, Section 6.6].

Lemma 6.82. *There are at most*

$$(4D_{SoS})^{|V(\tau_P)|-\frac{|U_{\tau_P}|+|V_{\tau_P}|}{2}+k}(3D_V)^k$$

non-equivalent intersection patterns $P \in \mathcal{P}^{mid}_{\gamma_j,\ldots,\gamma'_j}$, which have exactly k intersections.

Lemma 6.83. *For any intersection pattern P which has exactly k intersections,*

$$N_P(\tau_P) \leq \frac{D_{SoS}^{2k+|V(\tau_P)|-\frac{|U_{\tau_P}|+|V_{\tau_P}|}{2}}}{|U_{\tau_P} \cap V_{\tau_P}|!} |\operatorname{Aut}(\tau_P)|$$

6.4.6 Counting shapes with the tail bound function $c(\alpha)$

We have collected the unsymmetrized ribbons into symmetrized shapes, and we must upper bound the number of possible shapes σ, γ_i and τ that are summed over. We will do this by introducing a "tail bound function" $c(\alpha)$, such that $\sum_{\text{nontrivial permissible shapes } \alpha} 1/c(\alpha) \ll 1$. The function will be of the form $c(\alpha) = f(\alpha)^{|V(\alpha)\setminus(U_\alpha \cap V_\alpha)|} g(|E(\alpha)|)$. That is, we need a vertex decay of $f(\alpha)$ per vertex not in $U_\alpha \cap V_\alpha$, as well as a decay from conditioning when the shape α has too many edges.

Definition 6.84. *Let C' be an upper bound on the sparsity of shapes in Lemma 6.81. Given a permissible shape α such that $|V(\alpha) \setminus (U_\alpha \cap V_\alpha)| = v \geq 1$, there are e edges in $E(\alpha)$ which are not incident to a vertex in $U_\alpha \cap V_\alpha$, and $\alpha \setminus (U_\alpha \cap V_\alpha)$ has k connected components which are disconnected from $U_\alpha \cup V_\alpha$, we define $c(\alpha)$ by:*

1. *If $v \leq CD_{\text{SoS}}$ and $e \leq C'v$ then*

$$c(\alpha) = 40(2CD_{\text{SoS}}^{4C'+2})^{|V(\alpha)| - \frac{|U_\alpha|+|V_\alpha|}{2}}$$

2. *If $v \leq CD_{\text{SoS}}$ and $e > C'v$ then*

$$c(\alpha) = 40(2CD_{\text{SoS}}^{4C'+2})^{|V(\alpha)| - \frac{|U_\alpha|+|V_\alpha|}{2}} (2C^2 D_{\text{SoS}}^2)^{(e-C'v)}$$

3. *If $v > CD_{\text{SoS}}$ and $e \leq v - k + CD_{\text{SoS}}$ then*

$$c(\alpha) = 40(16CD_{\text{SoS}}^4)^{|V(\alpha)| - \frac{|U_\alpha|+|V_\alpha|}{2}}$$

4. *If $v > CD_{\text{SoS}}$ and $e > v - k + CD_{\text{SoS}}$ then*

$$c(\alpha) = 40(16CD_{\text{SoS}}^4)^{|V(\alpha)| - \frac{|U_\alpha|+|V_\alpha|}{2}} (8D_V)^{e+k-v-CD_{\text{SoS}}}$$

If α is a trivial shape then we define $c(\alpha) = 1$.

Lemma 6.85 ([JPR+21, Section 6.7])**.**

$$\sum_{\substack{\alpha \in \mathcal{S}: \\ \textit{non-trivial,} \\ \textit{permissible}}} \frac{1}{c(\alpha)} \leq \frac{1}{10}.$$

6.4.7 Statement of main lemmas

Now that we have wrangled the moment matrix into the correct form, we will show that all the error terms are small. Recall the expression for the moment matrix from

Lemma 6.81:

$$\Pi^{1/2}\left(\sum_{\substack{\text{sparse}\\ \text{permissible}\\ \sigma\in\mathcal{L}}}\lambda'_{\sigma}\frac{M_{\sigma}}{|\operatorname{Aut}(\sigma)|}\right)\cdot$$

$$\left(\sum_{j=0}^{2D_{\mathrm{SoS}}}(-1)^j\sum_{\substack{\text{sparse}\\ \text{permissible}\\ \gamma_j,\ldots,\gamma_1,\tau,\gamma'_1,\ldots,\gamma'_j}}\sum_{\substack{\text{nonequivalent}\\ P\in\mathcal{P}^{mid}_{\gamma_j,\ldots,\gamma_1,\tau,\gamma'_1,\ldots,\gamma'_j}}}N_P(\tau_P)\lambda'_{\gamma_j\circ\cdots\circ\gamma_1\circ\tau\circ\gamma_1'^{\intercal}\circ\cdots\circ\gamma_j'^{\intercal}}\frac{M_{\tau_P}}{|\operatorname{Aut}(\tau_P)|}\right)\cdot$$

$$\left(\sum_{\substack{\text{sparse}\\ \text{permissible}\\ \sigma\in\mathcal{L}}}\lambda'_{\sigma}\frac{M_{\sigma}}{|\operatorname{Aut}(\sigma)|}\right)^{\intercal}\Pi^{1/2}$$

$$+\,\Pi^{1/2}\cdot\text{truncation error}\cdot\Pi^{1/2}$$

The following lemmas, for which the intuition was given in Section 6.2.4, are sufficient to prove Theorem 6.3.

Lemma 6.86. *(Non-trivial Middle Shapes) For all sparse permissible $\tau\in\mathcal{M}$ such that $|V(\tau)|>\frac{|U_\tau|+|V_\tau|}{2}$ and $|E_{mid}(\tau)|-|V(\tau)|\le CD_{\mathrm{SoS}}$,*

$$\lambda'_{\tau}\frac{\|M_\tau\|}{|\operatorname{Aut}(\tau)|}\le\frac{1}{c(\tau)}.$$

Lemma 6.87. *(Intersection Terms) For all $j\ge 1$ and sparse permissible $\gamma_j,\ldots,\tau,\ldots,\gamma'_j$ such that for each shape $|E_{mid}(\alpha)|-|V(\alpha)|\le CD_{\mathrm{SoS}}$,*

$$\sum_{\substack{\textit{nonequivalent}\\ P\in\mathcal{P}^{mid}_{\gamma_j,\ldots,\gamma'_j}}}N_P(\tau_P)\lambda'_{\gamma_j\circ\cdots\circ\gamma_j'^{\intercal}}\frac{\|M_{\tau_P}\|}{|\operatorname{Aut}(\tau_P)|}\le\frac{1}{c(\tau)\prod_{i=1}^{j}c(\gamma_i)c(\gamma'_i)}.$$

Lemma 6.88. *(Truncation Error)*

$$\textit{truncation error}\preceq n^{-\Omega(CD_{\mathrm{SoS}})}\pi.$$

Lemma 6.89. *(Sum of left shapes is well-conditioned)*

$$\left(\sum_{\substack{\textit{sparse,}\\ \textit{permissible}\\ \sigma\in\mathcal{L}}} \lambda'_\sigma \frac{M_\sigma}{|\operatorname{Aut}(\sigma)|}\right)\left(\sum_{\substack{\textit{sparse,}\\ \textit{permissible}\\ \sigma\in\mathcal{L}}} \lambda'_\sigma \frac{M_\sigma}{|\operatorname{Aut}(\sigma)|}\right)^{\mathsf{T}} \succeq n^{-O(D_{\mathrm{SoS}})}\pi$$

Proof of Theorem 6.3 assuming Lemma 6.86, Lemma 6.87, Lemma 6.88, Lemma 6.89. For ease of exposition we omit the automorphism groups. In the approximate PSD decomposition, the term $j = 0$ can be broken up into the leading term π and remaining nontrivial middle shapes,

$$\begin{aligned}\sum_{\substack{\text{sparse,}\\ \text{permissible}\\ \tau\in\mathcal{M}}} \lambda'_\tau M_\tau &= \sum_{\substack{\text{permissible}\\ \tau\in\mathcal{M}:\\ U_\tau=V_\tau=V(\tau)}} \lambda_\tau M_\tau &&+ \sum_{\substack{\text{sparse,}\\ \text{permissible}\\ \tau\in\mathcal{M}:\\ |V(\tau)|>\frac{|U_\tau|+|V_\tau|}{2}}} \lambda'_\tau M_\tau\\ &= \pi &&+ \sum_{\substack{\text{sparse,}\\ \text{permissible}\\ \tau\in\mathcal{M}:\\ |V(\tau)|>\frac{|U_\tau|+|V_\tau|}{2}}} \lambda'_\tau M_\tau.\end{aligned}$$

Using Lemma 6.60, the middle shapes ($j = 0$) and intersection terms ($j \geq 1$) are

$$\pi\Bigg(\operatorname{Id} + \sum_{\substack{\text{sparse,}\\ \text{permissible}\\ \tau\in\mathcal{M}:\\ |V(\tau)|>\frac{|U_\tau|+|V_\tau|}{2}}} \lambda'_\tau M_\tau + \sum_{j=1}^{2D_{\mathrm{SoS}}} (-1)^j \sum_{\substack{\text{sparse,}\\ \text{permissible}\\ \gamma_j,\ldots,\gamma'_j}} \sum_{\substack{\text{nonequivalent:}\\ P\in\mathcal{P}^{mid}_{\gamma_j,\ldots,\gamma'_j}}} N_P(\tau_P)\lambda'_{\gamma_j\circ\cdots\circ\gamma_j'^{\mathsf{T}}} M_{\tau_P}\Bigg).$$

By Lemma 6.86 and Lemma 6.87, (summed with Lemma 6.85) this is at least $\Omega(1)\pi$.

Now plugging this into the PSD decomposition, we have:

$$\sum_{\alpha\in\mathcal{S}} \lambda_\alpha \frac{M_\alpha}{|\operatorname{Aut}(\alpha)|} \succeq \Pi^{1/2}\left(\sum_{\substack{\text{sparse}\\ \text{permissible}\\ \sigma\in\mathcal{L}}} \lambda'_\sigma M_\sigma\right)\Omega(1)\pi\left(\sum_{\substack{\text{sparse}\\ \text{permissible}\\ \sigma\in\mathcal{L}}} \lambda'_\sigma M_\sigma\right)^{\mathsf{T}} \Pi^{1/2} + \text{truncation error}$$

By Lemma 6.60 again,

$$= \Omega(1)\Pi^{1/2}\left(\sum_{\substack{\text{sparse}\\ \text{permissible}\\ \sigma\in\mathcal{L}}}\lambda'_\sigma M_\sigma\right)\left(\sum_{\substack{\text{sparse}\\ \text{permissible}\\ \sigma\in\mathcal{L}}}\lambda'_\sigma M_\sigma\right)^{\mathsf{T}}\Pi^{1/2} + \text{truncation error}$$

By Lemma 6.89, Lemma 6.88, and taking C sufficiently large,

$$\begin{aligned}&\succeq n^{-O(D_{\mathrm{SoS}})}\Pi - n^{-\Omega(CD_{\mathrm{SoS}})}\Pi\\ &\succeq 0\end{aligned}$$

□

6.4.8 Conditioning II: Frobenius norm trick

Shapes with a large number of excess edges should be handled using their Frobenius norm. The Frobenius norm trick improves on the first moment method, Proposition 6.68, so it could be used to also handle $O(1)$-sparse subgraphs in this work. On the other hand, the forbidden subgraph method can handle forbidden graphs beyond the first moment method, which may lead to improvements for smaller d.

There are two parts to the Frobenius norm trick. First, we compute the Frobenius norm of a graph matrix.

Lemma 6.90. *For all proper shapes α,*

$$\mathbb{E}[\operatorname{tr}(M_\alpha M_\alpha^{\mathsf{T}})] \le |\operatorname{Aut}(\alpha)| n^{|V(\alpha)|+|I_\alpha|}.$$

Proof. Any term contributing to $\mathbb{E}\left[\operatorname{tr}(M_\alpha M_\alpha^{\mathsf{T}})\right]$ is a labeling of α and α^{T} s.t. every edge appears exactly twice. After choosing the labels for α in $n^{|V(\alpha)|}$ ways, the labeling of α^{T} gives an isomorphism between α and α^{T} (except for isolated vertices, which give an extra factor of n each). □

For proper shapes α, this gives a norm bound independent of the number of edges! MVS with a large number of induced edges can't hurt us. We have

$$\lambda_\alpha \|M_\alpha\| \le \left(\frac{k}{n}\right)^{|V(\alpha)| - \frac{|U_\alpha|+|V_\alpha|}{2}} \sqrt{p}^{|E(\alpha)|}\sqrt{n}^{|V(\alpha)|}.$$

The number of edges could be smaller than the number of vertices, but only by D_{SoS} (since everything must be connected to $U_\alpha \cup V_\alpha$). If there are at least CD_{SoS} excess edges, then the norm is small enough to sum.

The Frobenius norm is small for too-dense subgraphs of G. The second part of the

Frobenius norm trick allows us to use these bounds with high probability instead of in expectation, which we can do using Markov's inequality once instead of using Markov's inequality on each one.

Lemma 6.91. *For any random matrices $M_1, \ldots, M_k$,*

$$\mathbb{E}\left[\operatorname{tr}\left(\left(\sum_{i=1}^{k} M_i\right)\left(\sum_{i=1}^{k} M_i\right)^{\intercal}\right)\right] \leq \left(\sum_{i=1}^{k} \sqrt{\mathbb{E}\left[\operatorname{tr}\left(M_i {M_i}^{\intercal}\right)\right]}\right)^2$$

Proof. Observe that by Cauchy-Schwarz,

$$\begin{aligned}\mathbb{E}\left[\operatorname{tr}(M_i {M_j}^{\intercal})\right] &= \mathbb{E}\left[\sum_{a,b} (M_i)_{ab}(M_j)_{ab}\right] \\ &\leq \sqrt{\mathbb{E}\left[\sum_{a,b} (M_i)_{ab}^2\right]}\sqrt{\mathbb{E}\left[\sum_{a,b} (M_j)_{ab}^2\right]} \\ &= \sqrt{\mathbb{E}\left[\operatorname{tr}\left(M_i {M_i}^{\intercal}\right)\right]}\sqrt{\mathbb{E}\left[\operatorname{tr}\left(M_j {M_j}^{\intercal}\right)\right]}\end{aligned}$$

Applying this inequality for all $i, j \in [k]$ gives the result. □

6.4.9 Norm bounds

For improper shapes, we linearize them. Using orthonormality of the p-biased Fourier character, we may linearize the character by:

$$\chi^k = \mathop{\mathbb{E}}_{x \sim \text{Bernoulli}(p)}[\chi^k(x)] + \mathop{\mathbb{E}}_{x \sim \text{Bernoulli}(p)}[\chi^{k+1}(x)] \cdot \chi\,.$$

Proposition 6.92. *For all $k \in \mathbb{N}$ and all $p \leq \frac{1}{2}$,*

$$\left|\mathop{\mathbb{E}}_{x \sim \text{Bernoulli}(p)}[\chi^k(x)]\right| \leq \left(\sqrt{\frac{1-p}{p}}\right)^{k-2} \quad \textit{and} \quad \left|\mathop{\mathbb{E}}_{x \sim \text{Bernoulli}(p)}\left[\chi^k(x)\mathbb{1}_{x=0}\right]\right| \leq \left(\sqrt{\frac{1-p}{p}}\right)^{k-2}.$$

Proof.

$$\begin{aligned}\mathop{\mathbb{E}}_{x\sim\text{Bernoulli}(p)}[\chi^k(x)] &= p\cdot\sqrt{\frac{1-p}{p}}^{k} + (1-p)\cdot\left(-\sqrt{\frac{p}{1-p}}\right)^{k}\\ &= (1-p)\cdot\sqrt{\frac{1-p}{p}}^{k-2} + p\cdot\left(-\sqrt{\frac{p}{1-p}}\right)^{k-2}\\ &\le (1-p)\cdot\sqrt{\frac{1-p}{p}}^{k-2} + p\cdot\left(\sqrt{\frac{1-p}{p}}\right)^{k-2}\\ &= \sqrt{\frac{1-p}{p}}^{k-2}.\end{aligned}$$

The second inequality follows similarly.

□

Proposition 6.93. *For a shape α,*

$$M_\alpha = \sum_{\textit{linearizations } \beta \textit{ of } \alpha}\left(\sqrt{\frac{1-p}{p}}\right)^{|E(\alpha)|-|E(\beta)|-2|E_{phantom}(\beta)|} M_\beta.$$

Proof. Each Fourier character can be linearized as $\chi_e^k = \mathbb{E}[\chi_e^k] + \mathbb{E}[\chi_e^{k+1}]\chi_e$. Then use Proposition 6.92. □

Theorem 6.94. *If $D_V \ge \lceil D_{\text{SoS}}\ln(n)\rceil$ then for any shape α (including improper shapes, shapes with missing edge indicators, and shapes with quasi-missing edge indicators), taking M_α to be the graph matrix where the rows and columns are indexed by sets (i.e. the definition we use throughout the paper), with probability at least $1-\varepsilon'$ the following holds:*

$$\|M_\alpha\| \le 20\left(\frac{2^{|E(\alpha)|-|E_{no\ repetitions}(\alpha)|}}{\epsilon'}\right)^{\frac{1}{2D_V}} 2^{|E(\alpha)|-|E_{no\ repetitions}(\alpha)|}\sqrt{|U_\alpha|!|V_\alpha|!}$$
$$\max_{\beta,S}\left\{n^{\frac{|V(\alpha)|-|S|+|I_\beta|}{2}}(12D_V)^{|V(\alpha)|-\frac{|U_\alpha|+|V_\alpha|}{2}+|S|-\frac{|L_S|+|R_S|}{2}}\sqrt{\frac{1-p}{p}}^{|E(\alpha)|-|E(\beta)|-2|E_{phantom}(\beta)|}\left(3\sqrt{\frac{1-p}{p}}\right)^{|E(S)|}\right\}$$

where $E_{no\ repetitions}(\alpha)$ is the set (rather than the multi-set) of edges of α, β is a linearization of α, S is a separator of β, I_β is the set of vertices of β which are isolated, and $E_{phantom}(\beta)$ is the set of edges of α which are not in β. As a special case, when α is a proper shape with no isolated vertices (which may still contain missing edge indicators and quasi-missing edge indicators), with *probability at least $1-\varepsilon'$ the following holds:*

$$\|M_\alpha\| \leq 20\left(\frac{1}{\epsilon'}\right)^{\frac{1}{2D_V}} \sqrt{|U_\alpha|!|V_\alpha|!} \max_S \left\{ n^{\frac{|V(\alpha)|-|S|}{2}} (12D_V)^{|V(\alpha)|-\frac{|U_\alpha|+|V_\alpha|}{2}+|S|-\frac{|L_S|+|R_S|}{2}} \left(3\sqrt{\frac{1-p}{p}}\right)^{|E(S)|} \right\}$$

Proof. Omitted. See [JPR$^+$21, Section 6.10]. □

One more proposition is needed to handle the automorphism terms that arise when switching between ribbons/shapes/shapes indexed by ordered tuples,

Proposition 6.95. *For a permissible shape α of degree at most D_{SoS},*

$$\frac{\sqrt{|U_\alpha|!|V_\alpha|!}}{|\operatorname{Aut}(\alpha)|} \leq D_{\text{SoS}}^{|V(\alpha)\setminus(U_\alpha\cap V_\alpha)|/2+|V(\alpha)\setminus(U_\alpha\cup V_\alpha)|}$$

Proof. Write

$$\frac{\sqrt{|U_\alpha|!|V_\alpha|!}}{|\operatorname{Aut}(\alpha)|} = \frac{\sqrt{|U_\alpha|!|V_\alpha|!}}{|U_\alpha\cap V_\alpha|!} \cdot \frac{|U_\alpha\cap V_\alpha|!}{|\operatorname{Aut}(\alpha)|}$$

The first ratio is upper bounded by $D_{\text{SoS}}^{|V(\alpha)\setminus(U_\alpha\cap V_\alpha)|/2}$.

For the second ratio, in a permissible shape, $|\operatorname{Aut}(\alpha)| \geq (|U_\alpha\cap V_\alpha| - |V(\alpha)\setminus(U_\alpha\cup V_\alpha)|)!$. This is because in a permissible shape, the number of edges incident to $U_\alpha\cap V_\alpha$ is at most the number of connected components of $V(\alpha)\setminus(U_\alpha\cup V_\alpha)$, which is upper bounded by $|V(\alpha)\setminus(U_\alpha\cup V_\alpha)|$. The remaining vertices of $U\cap V$ that are not incident to an edge are completely interchangeable. Therefore this ratio is upper bounded by $D_{\text{SoS}}^{|V(\alpha)\setminus(U_\alpha\cup V_\alpha)|}$. □

6.4.10 Proof of main lemmas

Lemma 6.86: nontrivial middle shapes

Lemma 6.86. *(Non-trivial Middle Shapes) For all sparse permissible $\tau\in\mathcal{M}$ such that $|V(\tau)| > \frac{|U_\tau|+|V_\tau|}{2}$ and $|E_{mid}(\tau)| - |V(\tau)| \leq CD_{\text{SoS}}$,*

$$\lambda'_\tau \frac{\|M_\tau\|}{|\operatorname{Aut}(\tau)|} \leq \frac{1}{c(\tau)}.$$

Proof. The remaining "core" shapes are $\tau\in\mathcal{M}$ nontrivial, sparse, permissible with $|E(\alpha)| \leq |V(\alpha)| + CD_{\text{SoS}}$. For these shapes, the norm bound in Theorem 6.94 with $\varepsilon' = \frac{1}{nc(\tau)}$ holds whp, via Lemma 6.85.

$$\begin{aligned}
&\lambda'_\tau \frac{\|M_\tau\|}{|\operatorname{Aut}(\tau)|} \\
&\qquad \leq m(\tau) \left(\lambda_\tau + \sum_{U \supset \tau} \lambda_U c(\tau, U) \right) \cdot \\
&\qquad C \left(2^{|E(\tau)|} n c(\tau) \right)^{\frac{1}{2D_V}} 2^{|E(\tau)|} . \\
&\qquad \max_{\text{separator } S} \left\{ n^{\frac{|V(\alpha)|-|S|}{2}} (12D_V)^{|V(\alpha)| - \frac{|U_\alpha|+|V_\alpha|}{2} + |S| - \frac{|L_S|+|R_S|}{2}} \left(3\sqrt{\frac{1-p}{p}} \right)^{|E(S)|} \right\} \\
&\qquad \cdot \frac{\sqrt{|U_\tau|!|V_\tau|!}}{|\operatorname{Aut}(\tau)|}
\end{aligned}$$

We have:

$$\begin{aligned}
m(\tau) &\leq \left(\frac{1}{(1-p)^{2D_{\text{SoS}}}} \right)^{|V(\tau) \setminus (U_\tau \cap V_\tau)|} \leq 2^{|V(\tau) \setminus (U_\tau \cap V_\tau)|} && \text{(Lemma 6.63)} \\
\sum_{U \supset \tau} \lambda_U c(\tau, U) &= o(1) \lambda_\tau && \text{(Lemma 6.78)} \\
2^{|E(\alpha)|/2D_V} &\leq C && (\tau \text{ is sparse}) \\
n^{1/2D_V} &\leq 2 \\
c(\tau) &\leq C D_{\text{SoS}}^{c|V(\tau) \setminus (U_\tau \cap V_\tau)|} && \text{(Definition 6.84, } \tau \text{ is sparse)} \\
c(\tau)^{\frac{1}{2D_V}} &\leq D_{\text{SoS}}^c && \text{(From previous line)} \\
3^{|E(S)|} &\leq C^{|V(\tau) \setminus (U_\tau \cap V_\tau)|} && (\tau \text{ is sparse}) \\
\frac{\sqrt{|U_\tau|!|V_\tau|!}}{|\operatorname{Aut}(\tau)|} &\leq C D_{\text{SoS}}^{c|V(\tau) \setminus (U_\tau \cap V_\tau)|} && \text{(Proposition 6.95))}
\end{aligned}$$

Going through the charging argument for middle shapes in Proposition 6.18,

$$\lambda_\tau n^{\frac{|V(\alpha)|-|S|}{2}} \left(\sqrt{\frac{1-p}{p}} \right)^{|E(S)|} \leq \left(\frac{k\sqrt{d}}{n} \right)^{|V(\tau)| - \frac{|U_\tau|+|V_\tau|}{2}} \cdot \left(\frac{1}{\sqrt{d}} \right)^{|S| - \frac{|U_\tau|+|V_\tau|}{2}} .$$

Using $D_{\text{SoS}} \leq \frac{d^{1/2}}{\log n}$, we have $\sqrt{d} \geq D_V$ and the last term cancels $D_V^{|S| - \frac{|L_S|+|R_S|}{2}}$ from the norm bound after the following claim:

Claim 6.96. *For a middle shape τ and any vertex separator S, $|L_S| \geq |U_\tau|$ and $|R_S| \geq |V_\tau|$.*

Proof. L_S and R_S are both vertex separators of τ, and since τ is a middle shape, U_τ, V_τ are MVSs of τ. □

Putting it together, the vertex decay of $\frac{k\sqrt{d}}{n}$ on $|V(\tau)| - \frac{|U_\tau|+|V_\tau|}{2}$ needs to be $CD_V D_{\mathrm{SoS}}^c$ to overcome the combinatorial factors, as stated in Theorem 6.3.

$$\lambda'_\tau \frac{\|M_\tau\|}{|\mathrm{Aut}(\tau)|} \le \frac{1}{c(\tau)}.$$

□

Lemma 6.87: bounding intersection terms

We will need the following proposition.

Proposition 6.97. *For any C'-sparse permissible composable shapes $\alpha_1, \dots, \alpha_k$ and any intersection pattern P on $\alpha_1, \dots, \alpha_k$, letting τ_P be the resulting shape,*

$$|E(\tau_P)| \le (2C'+1) \sum_{i=1}^{k} \left(|V(\alpha_i)| - \frac{|U_{\alpha_i}| + |V_{\alpha_i}|}{2} \right)$$

Proof. Observe that since α_i is permissible, the only edges incident to $U_{\alpha_i} \cap V_{\alpha_i}$ are one for each connected component of $V(\alpha_i) \setminus (U_{\alpha_i} \cup V_{\alpha_i})$. Therefore, there are at most $|V(\alpha_i) \setminus (U_{\alpha_i} \cup V_{\alpha_i})|$ of these edges. Since α_i is C'-sparse,

$$|E(\alpha_i)| \le |V(\alpha_i) \setminus (U_{\alpha_i} \cup V_{\alpha_i})| + C'|V(\alpha) \setminus (U_{\alpha_i} \cap V_{\alpha_i})| \le (2C'+1) \left(|V(\alpha_i)| - \frac{|U_{\alpha_i}| + |V_{\alpha_i}|}{2} \right)$$

Since $|E(\tau_P)| = \sum_{i=1}^{k} |E(\alpha_i)|$, summing this equation over all $i \in [k]$ gives the result. □

Lemma 6.87. *(Intersection Terms) For all $j \ge 1$ and sparse permissible $\gamma_j, \dots, \tau, \dots, \gamma'_j$ such that for each shape $|E_{mid}(\alpha)| - |V(\alpha)| \le CD_{\mathrm{SoS}}$,*

$$\sum_{\substack{nonequivalent\\ P \in \mathcal{P}^{mid}_{\gamma_j, \dots, \gamma'_j}}} N_P(\tau_P) \lambda'_{\gamma_j \circ \dots \circ \gamma'^{\intercal}_j} \frac{\|M_{\tau_P}\|}{|\mathrm{Aut}(\tau_P)|} \le \frac{1}{c(\tau) \prod_{i=1}^{j} c(\gamma_i) c(\gamma'_i)}.$$

Proof. We index the intersecting shapes as $\alpha_i = \gamma_k, \dots, \gamma_1, \tau, \gamma'_1, \dots, \gamma'_k$. For each α_i, by Lemma 6.63 and Lemma 6.78,

$$\lambda'_{\alpha_i} \le 2^{|V(\alpha_i) \setminus (U_{\alpha_i} \cap V_{\alpha_i})|} \lambda_{\alpha_i}.$$

Let $p_\ell(\alpha_i)$ be the number of nonequivalent intersection patterns $P \in \mathcal{P}^{mid}_{\gamma_j, \dots, \gamma'_j}$ which have exactly ℓ intersections.

Apply the norm bound in Theorem 6.94 with $\varepsilon' = \frac{1}{n^C \prod_{i=1}^{2k+1} c(\alpha_i) p_k(\alpha_i)}$, which holds whp via Lemma 6.85.

$$\begin{aligned}
&\sum_{\text{non equiv. P}} N_p(\tau_P) \cdot \lambda_{\gamma_k \circ \cdots \circ \gamma_1 \circ \tau \circ \gamma_1^{\intercal} \circ \cdots \circ \gamma_k'^{\intercal}} \frac{\|M_{\tau_P}\|}{|\operatorname{Aut}(\tau_P)|} \\
\leq &\lambda_{\gamma_k \circ \cdots \circ \gamma_1 \circ \tau \circ \gamma_1^{\intercal} \circ \cdots \circ \gamma_k'^{\intercal}} \cdot 20 \left(2^{|E(\tau_P)|} n^C \prod_{i=1}^{2k+1} c(\alpha_i) p_\ell(\alpha_i) \right)^{\frac{1}{2D_V}} 2^{|E(\tau_P)|} \cdot \frac{\sqrt{|U_{\tau_P}|!|V_{\tau_P}|!}}{|\operatorname{Aut}(\tau_P)|} \cdot \\
&\max_{\beta, S} \left\{ n^{\frac{|V(\tau_P)| - |S| + |I_\beta|}{2}} (12 D_V)^{|V(\tau_P)| - \frac{|U_{\tau_P}| + |V_{\tau_P}|}{2} + |S| - \frac{|L_S| + |R_S|}{2}} \cdot \right. \\
&\left. \left(\sqrt{\frac{1-p}{p}} \right)^{|E(\tau_P)| - |E(\beta)| - 2|E_{phantom}(\beta)|} \left(3 \sqrt{\frac{1-p}{p}} \right)^{|E(S)|} \right\}.
\end{aligned}$$

Bounds:

$$\begin{aligned}
12^{|E(\tau_P)|} &\leq C^{\sum_{i=1}^{2k+1} |V(\alpha_i)| - \frac{|U_{\alpha_i}| + |V_{\alpha_i}|}{2}} && \text{(Proposition 6.97)} \\
n^{C/2D_V} &\leq 2 \\
c(\alpha_i) &\leq C (D_{\text{SoS}}^c)^{|V(\alpha_i) \setminus (U_{\alpha_i} \cap V_{\alpha_i})|} && \text{(Definition 6.84, } \alpha_i \text{ is sparse)} \\
c(\alpha_i)^{1/2D_V} &\leq D_{\text{SoS}}^c && \text{(From previous line)} \\
p_\ell(\alpha_i) &\leq (D_{SoS}^c)^{|V(\tau_P)| - \frac{|U_{\tau_P}| + |V_{\tau_P}|}{2}} D_V^\ell && \text{(Lemma 6.82)} \\
\sum_{\text{non equiv. P}} N_P(\tau_P) &\leq \frac{(D_{\text{SoS}}^c)^{\sum_{i=1}^{2k+1} |V(\alpha_i)| - \frac{|U_{\alpha_i}| + |V_{\alpha_i}|}{2}}}{|U_{\tau_P} \cap V_{\tau_P}|!} |\operatorname{Aut}(\tau_P)| && \text{(Lemma 6.83)} \\
\frac{\sqrt{|U_{\tau_P}|!|V_{\tau_P}|!}}{|\operatorname{Aut}(\tau_P)|} &\leq C (D_{\text{SoS}}^c)^{|V(\tau_P) \setminus (U_{\tau_P} \cap V_{\tau_P})|} && \text{(Proposition 6.95)}
\end{aligned}$$

Following the charging argument for intersection terms in Proposition 6.24,

$$\begin{aligned}
&\lambda_{\gamma_k \circ \cdots \circ \gamma_1 \circ \tau \circ \gamma_1^{\intercal} \circ \cdots \circ \gamma_k'^{\intercal}} n^{\frac{|V(\tau_P)| - |S| + |I_\beta|}{2}} \left(\sqrt{\frac{1-p}{p}} \right)^{|E(S)| + |E(\alpha)| - |E(\beta)| - 2|E_{phantom}(\beta)|} \\
\leq &\left(\frac{k\sqrt{d}}{n} \right)^{\sum_{i=1}^{2k+1} |V(\alpha_i)| - \frac{|U_{\alpha_i}| + |V_{\alpha_i}|}{2}} \cdot \left(\frac{1}{\sqrt{d}} \right)^{i_P - |I_\beta| + |S| - \frac{|U_\beta| + |V_\beta|}{2}}.
\end{aligned}$$

In Proposition 6.24 we used the intersection tradeoff lemma, Lemma 6.25, to prove that the exponent of the second term is nonnegative. In that proof, use the following claim to get a lower bound of $|S| - \frac{|L_S|+|R_S|}{2}$ on the exponent.

Claim 6.98. $\frac{|L_S|+|R_S|}{2} \geq |S_{\tau_P,min}|$.

Proof. Both L_S and R_S are vertex separators of β, hence they are larger than $S_{\tau_P,min}$, which is the smallest separator among all linearizations of τ_P. □

Since $\sqrt{d} \geq D_V$, the second term cancels $D_V^{|S|-\frac{|L_S|+|R_S|}{2}}$ in the norm bound.

The remaining factors are mostly easily seen to be under control by the vertex decay. One exception is the term D_V^{ℓ} in $p_\ell(\alpha_i)$, the count of nonequivalent intersection patterns. By combining this with $D_V^{|V(\tau_P)|-\frac{|U_{\tau_P}|+|V_{\tau_P}|}{2}}$ from the norm bound we get at most $D_V^{\sum_{i=1}^{2k+1}|V(\alpha_i)|-\frac{|U_{\alpha_i}|+|U_{\alpha_i}|}{2}}$, which is controlled by one factor of D_V in the vertex decay. In total, a vertex decay of $CD_V D_{\text{SoS}}^c$ per vertex is sufficient.

Since each additional level of intersections is non-trivial, the total number of vertex decay factors is at least k, which is sufficient to handle CD_{SoS}^c per shape in addition to per vertex. □

Lemma 6.88: truncation error for shapes with too many vertices

Omitted. See [JPR$^+$21, Section 6.11.3].

Lemma 6.88: truncation error for shapes with too many edges in one part

Omitted. See [JPR$^+$21, Section 6.11.4].

Lemma 6.89: sum of left shapes is well-conditioned

Omitted. See [JPR$^+$21, Section 6.11.5].

6.5 Open Problems

Several other problems on sparse graphs are conjectured to be hard for SoS and it is our hope that the techniques here can help prove that these problems are hard for SoS. These problems include MaxCut (see Conjecture 4.86), *k*-Coloring, and Densest-*k*-Subgraph. For MaxCut in particular, since there are no constraints other than booleanity of the variables it may be possible to truncate away dense shapes, which we could not do here due to the presence of independent set indicator functions.

Another direction for further research is to handle random graphs which are not Erdős-Rényi. Since the techniques here depend on graph matrix norms, one would hope that they generalize to distributions such as d-regular graphs for which low-degree polynomials are still concentrated. However, in the non-iid setting, it is not clear what the analogue of graph matrices should be used due to the lack of a Fourier basis that is friendly to work with.

The polynomial constraint "$\sum_{v\in V} x_v = k$" is not satisfied exactly by our pseudoexpectation operator. It's possible that techniques from [Pan21] can be used to fix this.

The parameters in this paper can likely be improved. One direction is to remove the final factor of $\log n$ from our bound. This would allow us to prove an SoS lower bound for the "ultrasparse regime" $d = O(1)$ rather than $d \geq \log^2 n$. This setting is interesting as there is a nontrivial algorithm that finds an independent set of half optimal size [GS14, RV17b]. Furthermore, this algorithm is local in a sense that we don't define here. It would be extremely interesting if this algorithm could be converted into a rounding algorithm for constant-degree SoS.

Another direction is to improve the dependence on D_{SoS}. While our bound has a $\frac{1}{poly(D_{\text{SoS}})}$ dependence on D_{SoS}, we conjecture that the dependence should actually be $(1-p)^{O(D_{\text{SoS}})}$. If so, this would provide strong evidence for the prevailing wisdom in parameterized complexity and proof complexity that a maximum independent set of size k requires $n^{\Omega(k)}$ time to find/certify (corresponding to SoS degree $\Omega(k)$).

CHAPTER 7
CODE UPPER BOUNDS

Now we will switch gears and study an application to *coding theory*. A fundamental question in coding theory is the maximum size of a binary code given a blocklength parameter n and a minimum distance parameter d_n. This value is typically denoted by $A_2(n, d_n)$. A particularly important regime occurs when $\lim_{n\to\infty} d_n/n = \delta$ for some absolute constant $\delta \in (0, 1/2)$. In this regime, $A_2(n, d_n)$ is known to grow exponentially in n. However, the precise rate of this exponential growth remains an elusive major open problem.

It is a folklore conjecture that known upper bounds on the rate are far from tight. The best known upper bound on the rate for distances $\delta \in (0.273, 1/2)$ is obtained by constructing solutions to the dual program of *Delsarte's linear program*, as done by McEliece, Rodemich, Rumsey and Welch [MRRW77]. However, since their proof in the 1970s, upper bounds on the rate have not improved; our goal is to improve this bound.

The idea that we pursue is to generalize Delsarte's linear program to a *hierarchy* of convex relaxations of the value $A_2(n, d_n)$, with one convex relaxation for each $\ell \in \mathbb{N}$. For $\ell = 1$, the convex relaxation equals Delsarte's linear program. As ℓ increases, the value of the convex relaxation becomes a better approximation to the true value $A_2(n, d_n)$. If we can prove an upper bound on the value of a convex relaxation, as McEliece, Rodemich, Rumsey, and Welch did for Delsarte's linear program, we have the same upper bound on $A_2(n, d_n)$.

The hierarchy is defined in Section 7.2. Via *symmetrization*, the hierarchy admits several equivalent formulations that we show in Section 7.3, where we also prove that it is weaker than the sum-of-squares hierarchy. Completeness of the hierarchy is proven in Section 7.4. Unfortunately we are not able to theoretically analyze the value of the hierarchy to improve upper bounds on code rates. In principle all one needs to do is construct solutions to the dual program; in Section 7.5 we study this dual program.

The hierarchy we define only gives improved upper bounds for the class of *linear codes*. We will define hierarchies for both linear and general codes, then see that in the general case, the hierarchy collapses to the first level (i.e. it has the same value as Delsarte's linear program). The contrast between completeness in the linear case and triviality in the general case surfaces a natural question: are optimum codes very far from being linear?

Bibliography. This chapter is the work [CJJ22]. Some ongoing progress by the same set of authors has been added: Section 7.3.6, an improved completeness proof in Section 7.4, and Section 7.5. We take a slightly different perspective here than in [CJJ22], preferring to emphasize Boolean Fourier analysis, convex programming, and the unsymmetrized formulation of the hierarchy, whereas the paper derives the symmetrized version of the hierarchy directly from generalized Kravchuk polynomials and MacWilliams identities. The paper [CJJ22] contains a generalization of the hierarchy to association schemes that is omitted here.

7.1 Preliminaries

7.1.1 Definitions

In this chapter we view the Boolean hypercube as $\mathbb{F}_2^n$.

Definition 7.1 (Code). *A (binary) code C of blocklength n is a subset of $\mathbb{F}_2^n$.*

Definition 7.2 (Linear code). *A linear code is a subspace of $\mathbb{F}_2^n$.*

Definition 7.3 (Hamming weight and distance). *For a word $x \in \mathbb{F}_2^n$, we denote by $|x| := |\{i \in [n] \mid x_i \neq 0\}|$ its Hamming weight. Given two words $x, y \in \mathbb{F}_2^n$, we denote by $\Delta(x,y) := |x-y|$ their Hamming distance.*

Definition 7.4 (Distance of a code). *The (minimum) distance of C is defined by $\Delta(C) := \min\{\Delta(x,y) \mid x, y \in C \wedge x \neq y\}$.*

Definition 7.5 (Rate of a code). *The rate of C is defined by $r(C) := \log_2(|C|)/n$.*

In coding theory, we are usually interested in the tradeoff between rate and distance. As the rate increases, the code packs more and more codewords into $\mathbb{F}_2^n$. We would like these codewords to pairwise remain as far apart as possible.

Definition 7.6 ($A_2(n,d)$). *The maximum size of a code of blocklength n and minimum distance at least d is defined as*

$$A_2(n,d) := \max\{|C| \mid C \subseteq \mathbb{F}_2^n, \Delta(C) \geq d\}.$$

It is often convenient to denote the asymptotic basis of this growth as $2^{R_2(\delta)}$, where the rate $R_2(\delta)$ is defined as:

Definition 7.7 (Asymptotic rate). *We denote the asymptotic rate of codes of relative distance at least δ as*

$$R_2(\delta) := \limsup_{n \to \infty} \frac{1}{n} \log_2 \left(A_2(n, \lfloor \delta n \rfloor)\right).$$

Note that a code of distance at least d can alternatively be viewed as an independent set in the *Hamming cube graph of distance less than d, $H_{n,d}$,* which is defined by:

Definition 7.8. $V(H_{n,d}) := \mathbb{F}_2^n, E(H_{n,d}) := \{\{x,y\} \in \binom{\mathbb{F}_2^n}{2} \mid \Delta(x,y) \leq d-1\}$.

Definition 7.9 ($A_2^{\mathrm{Lin}}(n,d)$ and $R_2^{\mathrm{Lin}}(\delta)$). *We define $A_2^{\mathrm{Lin}}(n,d)$ and $R_2^{\mathrm{Lin}}(\delta)$ for linear codes in an analogous way, by further requiring the code C to be linear.*

Definition 7.10 (Dual code). *Given a linear code $C \subseteq \mathbb{F}_2^n$, the dual code of C is defined as $C^{\perp} := \{x \in \mathbb{F}_2^n \mid \forall y \in C, \chi_x(y) = 1\}$.*

The Fourier transform of the indicator of a linear code maps it to a multiple of the indicator of its dual code in the following way.

Fact 7.11. *If $C \subseteq \mathbb{F}_2^n$ is a linear code and $\mathbb{1}_C$ is its indicator function, then $\widehat{\mathbb{1}_C} = |C| \cdot \mathbb{1}_{C^\perp}/2^n = \mathbb{1}_{C^\perp}/|C^\perp|$.*

7.1.2 Problem background

Graph-theoretic interpretation Our goal is to understand the exponential growth rate $R_2(\delta)$ of the maximum code size $A_2(n,d)$ where $d \approx \delta n$. An equivalent way of defining $A_2(n,d)$ is as the independence number of the graph $H_{n,d}$ whose vertex set is $V(H_{n,d}) := \mathbb{F}_2^n$ and two vertices $x, y \in V(H_{n,d})$ are adjacent if and only if their Hamming distance $\Delta(x,y)$ lies in $\{1,\ldots,d-1\}$. Note that there is a one-to-one correspondence between independent sets in this graph and binary codes of blocklength n and minimum distance d. Most of the literature about $A_2(n,d)$ takes advantage of this graph-theoretic interpretation.

A lower bound on $A_2(n,d)$ follows from the trivial degree bound on the independence number of a graph, namely, $\alpha(H_{n,d}) \geq |V(H_{n,d})|/(\deg(H_{n,d})+1)$, which gives $\alpha(H_{n,d}) \geq 2^{(1-h_2(d/n)+o(1))n}$ where h_2 is the binary entropy function. First discovered by Gilbert [Gil52] and later generalized to linear codes by Varshamov [Var57], this existential bound is now known as the *Gilbert–Varshamov (GV) bound*. Observe that the GV bound readily implies that $R_2(\delta) \geq 1 - h_2(\delta)$. Despite its simplicity, this bound remains the best (existential) lower bound on $R_2(\delta)$.

A generic upper bound on the independence number of a graph is the Lovász ϑ function, which is a semidefinite programming relaxation for $\alpha(H_{n,d})$. The Lovász ϑ function also has the same value as the degree-2 sum-of-squares/Lasserre relaxation of $\alpha(H_{n,d})$. It turns out that this is essentially equal to the *Delsarte linear program* and it gives the strongest known upper bound on $A_2(n,d)$. However, the upper bound is known not to match the GV bound. Thus, it is natural to consider stronger convex relaxations for $A_2(n,d)$.

Viewed from this angle, the general problem can be seen as studying convex relaxations for independent set on a particular instance. The instance itself, $H_{n,d}$, is morally similar to the *noisy hypercube*. The noisy hypercube is a complete weighted graph whose vertex set is $\mathbb{F}_2^n$, and the weight of the edge between x, y is $\delta^{\Delta(x,y)}(1-\delta)^{n-\Delta(x,y)}$ for a parameter $\delta \in [0,1]$. Equivalently, the noisy hypercube is a Markov chain where a transition is made by flipping each bit with probability δ. Note that in the linear distance regime $d = \delta n, \delta \in (0, 1/2)$ that we are interested in, the degree of $H_{n,d}$ is polynomial in the graph size.

Delsarte linear programming method The strongest known upper bound on $A_2(n,d)$ is the *Delsarte linear program* [Del73]. The idea behind the method is to define a certain linear programming relaxation DelsarteLP(n,d) for $A_2(n,d)$. By virtue of being a relaxation, we have $A_2(n,d) \leq$ val(DelsarteLP(n,d)), and by proving an upper bound on val(DelsarteLP(n,d)), we have the same upper bound for $A_2(n,d)$. By virtue of being convex, the value of the program can be upper bounded by exhibiting any feasible dual solution. Using the dual linear program, McEliece, Rodemich, Rumsey, and

Welch [MRRW77] proved an upper bound on $\text{val}(\text{DelsarteLP}(n,d))$. Their "first linear programming bound" showed that $R_2(\delta) \leq h_2(1/2 - \sqrt{\delta(1-\delta)})$ by constructing a feasible dual solution using properties of the Kravchuk polynomials.[1]

There is a known gap between the value of Delsarte's linear program and the GV bound [NS05]. In particular when $\delta = 1/2 - \epsilon$, Delsarte's linear program does not yield an upper bound tighter than $R_2(1/2-\epsilon) \leq \Theta(\epsilon^2 \log(1/\epsilon))$ (this matches the analysis of [MRRW77]), whereas the GV bound establishes a lower bound of $R_2(1/2-\epsilon) \geq \Omega(\epsilon^2)$. There are no known improvements to these bounds even for the important class of *linear* codes. If the GV bound is indeed tight (i.e., a random construction is essentially optimal), then analyzing DelsarteLP is not sufficient to prove it.

As shown by Schrijver [Sch79], the Delsarte linear program is equal to the Lovász ϑ function (with a small modification, namely that Delsarte includes additional non-negativity constraints on the entries of the variable matrix). In fact, by *symmetrizing* the convex program for the ϑ' function, one obtains the Delsarte linear program exactly. Symmetrization is an important idea, as it reduces the exponential-in-n-size program for ϑ' on $H_{n,d}$ to a $\text{poly}(n)$-size program, which is more feasible to run and analyze. Furthermore, symmetrization reduces the positive semidefiniteness constraint to a system of linear inequalities, thus turning the semidefinite program for ϑ' into a linear program.

Although quantitatively the McEliece et al [MRRW77] upper bound on $R_2(\delta)$ has not improved, our qualitative understanding of this upper bound is now substantially better. Navon and Samorodnitsky [NS05] showed that the analysis of the Delsarte LP by McEliece et al is tight up to lower-order terms. Friedman and Tillich [FT05] designed generalized Alon–Boppana theorems in order to bound the size of linear binary codes. Inspired by Friedman and Tillich, Navon and Samorodnitsky [NS09] rederived the McEliece et al bound on $R_2(\delta)$ for general codes using a more intuitive proof based on Fourier analysis. Despite a seemingly different language, the proof in [NS09] also yields feasible solutions to the dual of Delsarte's LP as in [MRRW77]. More recently, Samorodnitsky [Sam21] gave yet a new interpretation of the McEliece et al upper bound and conjectured interesting hypercontractivity inequalities towards improving the upper bound on $R_2(\delta)$.

Stronger convex relaxations Prior work has also suggested using stronger convex relaxations to improve the upper bound on $A_2(n,d)$. When run experimentally on small n, these programs have surpassed the Delsarte linear program (small n are important for the real-world implementation of error-correcting codes). However, theoretically analyzing these larger relaxations to improve the asymptotic rate $R_2(\delta)$ seems challenging and has remained out of reach.

In Delsarte's approach, only the distance between pairs of points is taken into account in the optimization. For this reason, Delsarte's approach is classified as a 2-point bound. Nonetheless, there is no reason to restrict oneself to just 2-point interactions. Schri-

[1] In the same work, McEliece et al also gave the best known bound for $\delta \in (0, 0.273)$ via a second family of linear programs. Since our techniques are more similar to their first linear programming bound, we restrict our attention to it in this discussion.

jver [Sch05] constructed a family of semi-definite programs (SDPs) for $A_2(n,d)$ designed to take into account the 3-point interactions. Extending Schrijver's result to a 4-point interaction bound, Gijswijt, Mittelmann and Schrijver [GMS12] gave another tighter family of SDPs for $A_2(n,d)$ (they also give a description of their hierarchy for arbitrary level ℓ). The sum-of-squares relaxation for $\alpha(H_{n,d})$ was proposed by Laurent [Lau07], building on de Klerk et al [dKPS07]. So far, we do not even know how to analyze Schrijver's program, which is the simplest of these and is weaker than degree-4 sum-of-squares. Giving a "block diagonalization" to symmetrize and reduce the size of the SDP matrix has been a technically challenging step in these works [Gij09]. An advantage of our hierarchy is that complete diagonalization is trivial (the equivalent SDP is symmetrized into a linear program).

The sum-of-squares hierarchy is *complete*, meaning that a sufficiently high level of the hierarchy exactly recovers the value $\alpha(H_{n,d})$, essentially by brute force. There is an implicit hope that low levels of the hierarchy will provide nontrivially improving bounds over the Delsarte LP. Contrast this with the alternative, in which the value of the hierarchy at low levels hovers around the value of the first level of the hierarchy (the Delsarte LP), then suddenly "shoots down" to the correct value once the level becomes very large. We proved that this alternative behavior occurs for independent set on a *random* graph in Chapter 6. However, we are tentatively hopeful that for $H_{n,d}$, low levels do provide an improvement. A further implicit hope is that the improvement is on an exponential scale, i.e. the upper bound on the rate $R_2(\delta)$ is strictly monotonic with the level of the hierarchy.

To summarize, on the one hand, we have a thorough theoretical understanding of techniques based on Delsarte's LP, but if the true value of $A_2(n,d)$ or $A_2^{\text{Lin}}(n,d)$ is closer to the GV bound, then these techniques fall short of providing tight bounds. On the other hand, we have ℓ-point bounds from stronger SDPs which experimentally seem stronger, but (apparently) no clue how to theoretically analyze them to bound $R_2(\delta)$ for general codes or linear codes.

7.2 Definition of the Hierarchy

In this section, we define our hierarchy of convex relaxations $\text{FourierLP}(n,d,\ell)$ where $\ell \in \mathbb{N}_+$ is the level of the hierarchy. Note that $\text{FourierLP}(n,d,\ell)$ is a relaxation for $A_2(n,d)^\ell$ instead of $A_2(n,d)$. Level $\ell = 1$ will correspond to the Delsarte linear program (LP). We will actually define two hierarchies, $\text{FourierLP}(n,d,\ell)$ and $\text{FourierLP}_{\text{Lin}}(n,d,\ell)$, the second being a stronger hierarchy that only applies to *linear* codes.

To show how the hierarchy structurally generalizes the Delsarte LP, we start by recalling one formulation of the Delsarte LP, to be more formally denoted by $\text{DelsarteLP}(n,d)$. An equivalent formulation of this linear program was first introduced by Delsarte [Del73], where it was obtained in greater generality from the theory of association schemes.

The variables of the linear program are a_x, $(x \in \mathbb{F}_2^n)$, where in a true solution which is a linear code, a_x is the $0/1$ for whether x is in the code. Recall that the Fourier transform of the indicator function of a linear code is the indicator of the dual code (Fact 7.11).

The Delsarte LP relaxes this, so that the Fourier transform of a_x is simply required to be nonnegative. The overall relaxation for $A_2^{\mathrm{Lin}}(n,d)$ is the following:

$$\boxed{\begin{array}{llll}
 & \text{Variables: } a_x & x \in \mathbb{F}_2^n & \\
\max & \sum_{x\in\mathbb{F}_2^n} a_x & & \\
\text{s.t.} & a_0 = 1 & & \text{(Normalization)} \\
 & a_x = 0 & |x| \in \{1,\dots,d-1\} & \text{(Distance constraints)} \\
 & \sum_{x\in\mathbb{F}_2^n} a_x\chi_\alpha(x) \geq 0 & \forall \alpha \in \mathbb{F}_2^n & \text{(Fourier coefficients)} \\
 & a_x \geq 0 & \forall x \in \mathbb{F}_2^n & \text{(Non-negativity).}
\end{array}}$$

As we have described it, DelsarteLP(n,d) relaxes the maximum size of a linear code (recall that for a linear code, having distance at least d is equivalent to having no words of Hamming weight 1 through $d-1$, as enforced by the distance constraints). Somewhat surprisingly, MacWilliams, Sloane and Goethals proved that val(DelsarteLP(n,d)) also upper bounds $A_2(n,d)$ for general codes [MSG72].

Our LP hierarchy for linear codes can be simply described as checking non-negativity of products of Fourier coefficients. Define the linear programming hierarchy FourierLP$_{\mathrm{Lin}}(n,d,\ell)$ by:

$$\boxed{\begin{array}{llll}
 & \text{Variables: } a_x & x \in (\mathbb{F}_2^n)^\ell & \\
\max & \sum_{x\in(\mathbb{F}_2^n)^\ell} a_x & & \\
\text{s.t.} & a_0 = 1 & & \text{(Normalization)} \\
 & a_{(x_1,\dots,x_\ell)} = 0 & \exists w \in \mathrm{span}(x_1,\dots,x_\ell).\ |w| \in [d-1] & \text{(Distance constraints)} \\
 & \sum_{x\in(\mathbb{F}_2^n)^\ell} a_x\chi_\alpha(x) \geq 0 & \forall \alpha \in (\mathbb{F}_2^n)^\ell & \text{(Fourier coefficients)} \\
 & a_x \geq 0 & \forall x \in (\mathbb{F}_2^n)^\ell & \text{(Non-negativity).}
\end{array}}$$

Lemma 7.12. *For every $n,\ell \in \mathbb{N}_+$ and $d \in \{0,1,\dots,n\}$,* val(FourierLP$_{\mathrm{Lin}}(n,d,\ell)$) $\geq A_2^{\mathrm{Lin}}(n,d)^\ell$.

Proof. Given a linear code C with distance d, a feasible solution with value $|C|^\ell$ is $a_{(x_1,\dots,x_\ell)} := \prod_{i=1}^\ell \mathbb{1}[x_i \in C]$. The Fourier coefficient constraints are satisfied because

$$\sum_{x\in(\mathbb{F}_2^n)^\ell} \prod_{i=1}^{\ell} \mathbb{1}[x_i \in C]\chi_{\alpha_i}(x_i) = 2^{n\ell} \prod_{i=1}^{\ell} \widehat{\mathbb{1}_C}(\alpha_i),$$

which are nonnegative by Fact 7.11. □

In this hierarchy, the distance constraints are "semantic linearity constraints" that are only valid if the code is linear. The corresponding hierarchy for non-linear codes $\text{FourierLP}(n, d, \ell)$ is defined as:

$$\boxed{\begin{array}{llll}
 & \text{Variables: } a_x & x \in (\mathbb{F}_2^n)^\ell & \\
\max & \displaystyle\sum_{x \in (\mathbb{F}_2^n)^\ell} a_x & & \\
\text{s.t.} & a_0 = 1 & & \text{(Normalization)} \\
 & a_{(x_1,\ldots,x_\ell)} = 0 & \exists i \in [\ell].\ |x_i| \in \{1, \ldots, d-1\} & \text{(Distance constraints)} \\
 & \displaystyle\sum_{x \in (\mathbb{F}_2^n)^\ell} a_x \chi_\alpha(x) \geq 0 & \forall \alpha \in (\mathbb{F}_2^n)^\ell & \text{(Fourier coefficients)} \\
 & a_x \geq 0 & \forall x \in (\mathbb{F}_2^n)^\ell & \text{(Non-negativity).}
\end{array}}$$

The fact that this is a sound relaxation of $A_2(n, d)^\ell$ is shown in [CJJ22, Proposition 3.12]. Briefly, for a non-linear code C with distance d, we can construct a feasible solution by letting

$$a_{(x_1,\ldots,x_\ell)} = \frac{1}{|C|^\ell} |\{y_1, \ldots, y_\ell, z_1, \ldots, z_\ell \in C : \forall i \in [\ell].\ y_i - z_i = x_i\}| .$$

When C is linear, this recovers the indicator function of the code.

We make a few remarks about these two hierarchies.

1. The number of variables and constraints in the hierarchy is exponential in n, making the hierarchy infeasible to compute. By symmetrizing the hierarchy in the next section, we will reduce it to an equivalent program with $\text{poly}(n)$ size for fixed ℓ (in the same way that the standard formulation of the Delsarte linear program may be obtained from the formulation we have given here).

2. After symmetrizing, the variables of the hierarchy are interpreted as counting the number of ℓ-tuples of codewords with a given "weight configuration". In the linear case, it counts the number of low-dimensional subspaces (dimension $\leq \ell$) in the code with a given "weight configuration". In this sense the hierarchy takes advantage of "ℓ-point interactions" of codewords.

3. The two hierarchies actually exhibit different behavior. The hierarchy $\text{FourierLP}_{\text{Lin}}(n, d, \ell)$ for linear codes gives tighter and tighter bounds on $A_2^{\text{Lin}}(n, d)$ as ℓ increases, and it is complete at level $\ell = n$. On the other hand, the $\text{FourierLP}(n, d, \ell)$ hierarchy *does not improve on the Delsarte LP*. $\text{FourierLP}(n, d, \ell)$ is a "tensor product" of $\text{FourierLP}(n, d, 1) = \text{DelsarteLP}(n, d)$, and therefore solutions easily "lift". There is not enough structure in a general code to add nontrivial distance constraints. Both of these results may be found in Section 7.4.1.

4. The hierarchy is weaker than sum-of-squares (Section 7.3.6).

5. The hierarchy is of linear programs rather than semidefinite programs. The dual program is relatively clean and similar to other duals in the literature, thus offering hope for a theoretical analysis (see Section 7.5).

We view the main contribution of the hierarchy as being sufficiently powerful to ensure completeness while still being sufficiently simple to remain a hierarchy of linear programs (as opposed to genuine SDPs such as sum-of-squares), and bearing enough similarities with the original Delsarte LP to be amenable to theoretical analysis.

7.3 Alternative Formulations of the Hierarchy

7.3.1 Symmetrization of convex programs

The program FourierLP can be symmetrized into a smaller, equivalent program. We will briefly describe the technique of symmetrizing convex programs, which is also described in the survey article by Vallentin [Val19].

The technique exploits the fact that convex relaxations for the independence number $\alpha(H_{n,d})$ of the Hamming cube graph $H_{n,d}$ of distance less than d are highly symmetric, that is, programs that are invariant under large permutation groups as defined below.

Definition 7.13 (Program invariance)**.** *Let $\mathcal{P}$ be a linear program with variables $(a_x)_{x\in X}$ for some set X. We say that $\mathcal{P}$ is invariant under a permutation σ of X if for all feasible solutions (a_x), the point $a \cdot \sigma$ defined by $(a\cdot\sigma)_x := a_{\sigma(x)}$ is also feasible, and the objective value is the same.*

Similarly, a semidefinite program $\mathcal{P}$ with variable $M \in \mathbb{R}^{X\times X}$ is invariant under σ if for all feasible M, the matrix $M\cdot\sigma$ defined by $(M\cdot\sigma)[x,y] := M[\sigma(x),\sigma(y)]$ is also feasible, and the objective value is the same.

The group of permutations of X under which $\mathcal{P}$ is invariant is called the automorphism group of $\mathcal{P}$ and is denoted $\operatorname{Aut}(\mathcal{P})$.

If the input of a program $\mathcal{P}$ is a graph G and the program only depends on the isomorphism class of G, then the program is invariant under the automorphism group $\operatorname{Aut}(G)$ of the graph G. For convex relaxations such as the Lovász ϑ-function or the sum-of-squares hierarchy, the variables of the program are indexed by tuples of vertices from G, and thus a case of interest is when $\operatorname{Aut}(G)$ acts diagonally on tuples of vertices.

By symmetrizing solutions, i.e., by averaging the values of the variables over the automorphism group $\operatorname{Aut}(\mathcal{P})$, we may assume that the solution has the same symmetry:

Fact 7.14. *For any $H \subseteq \operatorname{Aut}(\mathcal{P})$, the value $\operatorname{val}(\mathcal{P})$ equals the value of $\mathcal{P}$ with the additional constraints $\forall\sigma\in H, \forall x \in X, a_x = a_{\sigma(x)}$ (or $\forall\sigma\in H, \forall x,y\in X, M[x,y] = M[\sigma(x),\sigma(y)]$ for an SDP).*

A symmetrized solution is constant on each orbit of the group action on X or X^2. Therefore, the "effective" number of variables in the convex program is only the number of orbits, which may be significantly smaller than even $|V(G)|$.

For example, the graph $H_{n,d}$ has a large symmetry group:

Fact 7.15. *For $1 < d < n$, $\mathrm{Aut}(H_{n,d})$ is the hyperoctahedral group, which is the semidirect product $\mathbb{F}_2^n \rtimes S_n$ in which S_n permutes the coordinates and $\mathbb{F}_2^n$ applies a bit flip.*

Even though the hypercube has size 2^n and thus $|V(H_{n,\ell})^\ell| = 2^{n\ell}$, the number of orbits of the diagonal action of $\mathrm{Aut}(H_{n,d})$ on ℓ-tuples is only $\mathrm{poly}(n)$ for constant ℓ. For example, for $\ell = 4$, viewing the hypercube momentarily as $\{-1, +1\}^n$, the orbit of (x_1, x_2, x_3, x_4) essentially only depends on the angles between the vectors: it is determined by the seven numbers

$$\langle x_1, x_2 \rangle, \quad \langle x_1, x_3 \rangle, \quad \langle x_1, x_4 \rangle, \quad \langle x_2, x_3 \rangle, \quad \langle x_2, x_4 \rangle, \quad \langle x_3, x_4 \rangle, \quad \sum_{i=1}^{n} x_{1,i} x_{2,i} x_{3,i} x_{4,i}. \quad (7.1)$$

Since each of the numbers in (7.1) takes at most $n + 1$ values, the effective number of variables in the degree-4 sum-of-squares relaxation for $\alpha(H_{n,d})$ is at most $O(n^7)$. Thus, the search for an upper bound on an exponential-size object is reduced to a polynomial-size convex program! Of course, to actually run this in polynomial time, one also needs to show that this polynomial-size convex program can be computed in polynomial time (which rules out explicitly computing the original program then taking a quotient).

7.3.2 Higher-order Kravchuk polynomial hierarchy

The linear programs FourierLP and $\mathrm{FourierLP_{Lin}}$ are symmetric under the action of S_n on the coordinates of the codewords. We apply the symmetrization technique from the previous section to simplify FourierLP into a smaller program that uses *higher-order Kravchuk polynomials*.

The specific invariant action is: consider the natural right action of S_n on $\mathbb{F}_2^n$ given by $(x \cdot \sigma)_i := x_{\sigma(i)}$ ($x \in \mathbb{F}_2^n, \sigma \in S_n, i \in [n]$) and take the diagonal action of S_n on $(\mathbb{F}_2^n)^\ell$ given by

$$(x_1, \dots, x_\ell) \cdot \sigma := (x_1 \cdot \sigma, \dots, x_\ell \cdot \sigma) \quad ((x_1, \dots, x_\ell) \in (\mathbb{F}_2^n)^\ell, \sigma \in S_n).$$

By Fact 7.14 we may consider only solutions to the LP that are symmetrized over S_n, that is, we have $a_x = a_y$ for each $x, y \in (\mathbb{F}_2^n)^\ell$ in the same orbit of the S_n-action.

We associate to each orbit of S_n on $(\mathbb{F}_2^n)^\ell$ a combinatorial object that we call a *configuration*. In plain English, the symmetric difference configuration of an ℓ-tuple $(z_1, \dots, z_\ell) \in (\mathbb{F}_2^n)^\ell$ of words captures all Hamming weights of linear combinations of the words $(z_1, \dots, z_\ell)$.

Definition 7.16. *The symmetric difference configuration of the ℓ-tuple $(z_1, \dots, z_\ell) \in (\mathbb{F}_2^n)^\ell$ is* *the function* $\mathsf{Config}^{\Delta}_{n,\ell}(z_1,\ldots,z_\ell)\colon 2^{[\ell]} \to \{0,1,\ldots,n\}$ *defined by*

$$\mathsf{Config}^{\Delta}_{n,\ell}(z_1,\ldots,z_\ell)(J) := \left|\sum_{j\in J} z_j\right|,$$

for every $J \subseteq [\ell]$*. That is, the value of the function at* $J \subseteq [\ell]$ *is the Hamming weight of the linear combination* $\sum_{j\in J} z_j$*.*

Not every function $c : 2^{[\ell]} \to \{0,1,\ldots,n\}$ *corresponds to a valid configuration. Viewing* $\mathsf{Config}^{\Delta}_{n,\ell}$ *as a function* $(\mathbb{F}_2^n)^\ell \to \mathbb{R}^{2^{[\ell]}}$ *(i.e., a function from the space of* ℓ*-tuples of words to the space of functions* $2^{[\ell]} \to \mathbb{R}$*), we denote the set of (valid) symmetric difference configurations by* $\mathrm{im}(\mathsf{Config}^{\Delta}_{n,\ell})$*.*

Given a symmetric difference configuration $g \in \mathrm{im}(\mathsf{Config}^{\Delta}_{n,\ell})$*, we will also abuse notation and write* $(z_1,\ldots,z_\ell) \in g$ *to mean that* $(z_1,\ldots,z_\ell) \in (\mathbb{F}_2^n)^\ell$ *has configuration* $\mathsf{Config}^{\Delta}_{n,\ell}(z_1,\ldots,z_\ell) = g$*. In other words, this abuse of notation consists of thinking of a configuration as the set of all* ℓ*-tuples of words that have this configuration (i.e. in the orbit). We also let* $|g|$ *be the size of this orbit, i.e., the number of* ℓ*-tuples whose configuration is* g*.*

The trivial symmetric difference configuration is the constant 0 *function (denoted by* 0*), which is the symmetric configuration of the tuple* $(0,\ldots,0) \in (\mathbb{F}_2^n)^\ell$*.*

Lemma 7.17 ([CJJ22, Lemma 3.4])**.** *The following are equivalent for* $(x_1,\ldots,x_\ell),(y_1,\ldots,y_\ell) \in (\mathbb{F}_2^n)^\ell$*.*

1. $(x_1,\ldots,x_\ell)$ *and* $(y_1,\ldots,y_\ell)$ *are in the same* S_n*-orbit.*
2. $\mathsf{Config}^{\Delta}_{n,\ell}(x_1,\ldots,x_\ell) = \mathsf{Config}^{\Delta}_{n,\ell}(y_1,\ldots,y_\ell)$*.*

After symmetrizing FourierLP, since $a_{(x_1,\ldots,x_\ell)} = a_{(y_1,\ldots,y_\ell)}$ whenever $(x_1,\ldots,x_\ell)$ and $(y_1,\ldots,y_\ell)$ have the same configuration, the variables of the program may be considered as a_g for each configuration $g \in \mathrm{im}(\mathsf{Config}^{\Delta}_{n,\ell})$. The Fourier constraint can be re-expressed as:

$$\sum_{g\in \mathrm{im}(\mathsf{Config}^{\Delta}_{n,\ell})} a_g \cdot \left(\sum_{(x_1,\ldots,x_\ell)\in g} \prod_{j=1}^{\ell} \chi_{\alpha_j}(x_j)\right) \geq 0 \qquad \left(\forall \alpha \in (\mathbb{F}_2^n)^\ell\right).$$

The inner summation is a symmetrized Fourier character, which is naturally lifted to an orthogonal polynomial on the space of configurations. For example, in the case $\ell = 1$ i.e. the Delsarte linear program, a configuration is specified by a single Hamming weight $i \in \{0,1,\ldots,n\}$. The orthogonal polynomials that appear are the *Kravchuk polynomials* K_i (we also saw this in the background Section 1.4).

$$\sum_{\substack{x\in\mathbb{F}_2^n:\\ |x|=i}} \chi_\alpha(x) = K_i(|\alpha|).$$

For higher ℓ, we analogously term these functions *higher-order Kravchuk polynomials*.

Definition 7.18 (Higher-order Kravchuk polynomial). *Let $h \in \operatorname{im}(\mathsf{Config}^{\Delta}_{n,\ell})$ be a symmetric difference configuration. The higher-order Kravchuk polynomial indexed by h is the function $K_h \colon \operatorname{im}(\mathsf{Config}^{\Delta}_{n,\ell}) \to \mathbb{R}$ defined by*

$$\begin{aligned} K_h(g) &:= 2^{\ell n} \mathop{\mathbb{E}}_{(y_1,\ldots,y_\ell)\in(\mathbb{F}_2^n)^\ell}\left[\mathbb{1}_h(y_1,\ldots,y_\ell)\cdot\prod_{j=1}^{\ell}\chi_{x_j}(y_j)\right] \\ &= \sum_{(y_1,\ldots,y_\ell)\in h}\prod_{j=1}^{\ell}\chi_{x_j}(y_j), \end{aligned} \tag{7.2}$$

for every symmetric difference configuration $g \in \operatorname{im}(\mathsf{Config}^{\Delta}_{n,\ell})$, where $(x_1,\ldots,x_\ell) \in g$ is any ℓ-tuple of words with symmetric difference configuration g and $\mathbb{1}_h$ is the indicator function of the set of ℓ-tuples whose symmetric difference configuration is h ([CJJ22, Lemma 3.19] shows this is well-defined independent of the choice of $(x_1,\ldots,x_\ell)$).

We therefore arrive at the following symmetrized version of FourierLP (note that the variables are rescaled, see [CJJ22, Proposition 4.5] for details):

Definition 7.19. *For $n,\ell \in \mathbb{N}_+$ and $d \in \{0,1,\ldots,n\}$, we let* KravchukLP(n,d,ℓ) *be the following linear program.*

$$\boxed{\begin{array}{llll}
 & \text{Variables: } a_g & g \in \operatorname{im}(\mathsf{Config}^{\Delta}_{n,\ell}) & \\
\max & \displaystyle\sum_{g\in\operatorname{im}(\mathsf{Config}^{\Delta}_{n,\ell})} a_g & & \\
\text{s.t.} & a_0 = 1 & & \text{(Normalization)} \\
 & a_g = 0 & \forall g \in \mathsf{ForbConfig}(n,d,\ell) & \text{(Distance constraints)} \\
 & \displaystyle\sum_{g\in\operatorname{im}(\mathsf{Config}^{\Delta}_{n,\ell})} K_h(g)\cdot a_g \geq 0 & \forall h \in \operatorname{im}(\mathsf{Config}^{\Delta}_{n,\ell}) & \text{(MacWilliams inequalities)} \\
 & a_g \geq 0 & \forall g \in \operatorname{im}(\mathsf{Config}^{\Delta}_{n,\ell}) & \text{(Non-negativity)}
\end{array}}$$

where

$$\mathsf{ForbConfig}(n,d,\ell) := \{g \in \operatorname{im}(\mathsf{Config}^{\Delta}_{n,\ell}) \mid \exists j \in [\ell], g(\{j\}) \in \{1,\ldots,d-1\}\}.$$

We also define KravchukLP$_{\text{Lin}}(n,d,\ell)$ *as the linear program obtained by replacing the set* $\mathsf{ForbConfig}(n,d,\ell)$ *with*

$$\mathsf{ForbConfig}_{\text{Lin}}(n,d,\ell) := \{g \in \operatorname{im}(\mathsf{Config}^{\Delta}_{n,\ell}) \mid \exists J \subseteq [\ell], g(J) \in \{1,\ldots,d-1\}\}.$$

Proposition 7.20 ([CJJ22, Proposition 4.5]). *For every $n,\ell \in \mathbb{N}_+$ and every $d \in \{0,1\ldots,n\}$,* *we have*

$$\begin{aligned}\mathrm{val}(\mathrm{FourierLP}(n,d,\ell)) &= \mathrm{val}(\mathrm{KravchukLP}(n,d,\ell)),\\ \mathrm{val}(\mathrm{FourierLP}_{\mathrm{Lin}}(n,d,\ell)) &= \mathrm{val}(\mathrm{KravchukLP}_{\mathrm{Lin}}(n,d,\ell)).\end{aligned}$$

This formulation of the hierarchy nicely parallels all of the combinatorial intuition from the original formulation of the Delsarte linear program [Del73]. The original program was based on the *MacWilliams identities* [Mac63], which show that the *weight distribution* of the dual code $C^{\perp}$ is determined by the weight distribution of a linear code C (and there is a linear formula relating the two). Generalized MacWilliams identities also hold for the distribution of ℓ-configurations in C and $C^{\perp}$ (in general this is true for any *association scheme*). For this and more, we refer the reader to Section 3 of [CJJ22].

Even though the space $(\mathbb{F}_2^n)^{\ell}$ has exponential size in n (for a fixed ℓ), the next lemma says that the number of configurations is polynomial in n (for a fixed ℓ). Hence the size of KravchukLP(n,d,ℓ) is only poly(n) for fixed ℓ, which should be compared to the exponential size of FourierLP.[2]

Lemma 7.21 ([CJJ22, Lemma 3.3]). *We have*

$$\left|\mathrm{im}(\mathsf{Config}_{n,\ell}^{\Delta})\right| = \binom{n+2^{\ell}-1}{2^{\ell}-1}.$$

Remark 7.22. *The linear programs* FourierLP *and* FourierLP$_{\mathrm{Lin}}$ *are not invariant under the other automorphisms of the hypercube of the form $x \mapsto x+z$ ($z \in \mathbb{F}_2^n$), because of the normalization constraint and the distance constraints. It makes more sense to view the underlying space as $\mathbb{F}_2^n$ instead of the hypercube, which does not have the $\mathbb{F}_2^n$ automorphism because the origin is treated specially.*

7.3.3 Subspace symmetrized hierarchy

Continuing the symmetrization from the previous section, there is actually more symmetry in the program FourierLP than just S_n. In the case of the program for non-linear codes, there is a symmetry under the right action of S_{ℓ} on $(\mathbb{F}_2^n)^{\ell}$ that permutes the *words* $x_1,\ldots,x_{\ell}$, that is, we have $(x_1,\ldots,x_{\ell})\cdot\tau := (x_{\tau(1)},\ldots,x_{\tau(\ell)})$ ($(x_1,\ldots,x_{\ell}) \in (\mathbb{F}_2^n)^{\ell}$, $\tau \in S_{\ell}$). In the case of the program for linear codes, we have symmetry under the action of $\mathrm{GL}_{\ell}(\mathbb{F}_2)$ that applies a basis change to $(x_1,\ldots,x_{\ell})$, that is, it is given by

$$(A\cdot x)_i := \sum_{j\in[\ell]} A[i,j]\cdot x_j \in \mathbb{F}_2^n \qquad (A \in \mathrm{GL}_{\ell}(\mathbb{F}_2), x \in (\mathbb{F}_2^n)^{\ell}, i \in [\ell]).$$

[2] Computing the higher-order Kravchuk polynomials in poly(n) time needs to be done using dynamic programming via a recursion formula [CJJ22, Lemma 3.19 or Lemma 3.20].

The distance constraints are evidently invariant under this action as it does not change the linear subspace spanned by $(x_1, \dots, x_\ell)$. The Fourier constraints are invariant since

$$\chi_\alpha(A \cdot x) = \chi_{A^\top \cdot \alpha}(x)$$

for every $x, \alpha \in \mathbb{F}_2^\ell$.

Note that the actions of $\mathrm{GL}_\ell(\mathbb{F}_2)$ and S_n commute with each other and thus induce an action of the direct product $\mathrm{GL}_\ell(\mathbb{F}) \times S_n$ on $(\mathbb{F}_2^n)^\ell$.

If we symmetrize the program $\mathrm{FourierLP}_{\mathrm{Lin}}(n, d, \ell)$ over $\mathrm{GL}_\ell(\mathbb{F}_2)$ and not over S_n, we obtain the following program with a variable for each subspace of $\mathbb{F}_2^n$ of dimension at most ℓ. (In just a moment, we will fully symmetrize over both actions, although it is not as easy to interpret the resulting orbits.)

Definition 7.23 (Grassmannian). *Let* $\mathrm{Gr}(\leq\ell, \mathbb{F}_2^n)$ *be the set of subspaces of* $\mathbb{F}_2^n$ *with dimension at most* ℓ. *Observe that* $\mathrm{Gr}(\leq\ell, \mathbb{F}_2^n)$ *describes the set of orbits of the action of* $\mathrm{GL}_\ell(\mathbb{F}_2)$ *on* $(\mathbb{F}_2^n)^\ell$.

Definition 7.24 (Subspace higher-order Kravchuks). *For* $U \in \mathrm{Gr}(\leq\ell, \mathbb{F}_2^n)$, *define:*

$$K_U^\ell : \mathrm{Gr}(\leq\ell, \mathbb{F}_2^n) \to \mathbb{R}$$
$$K_U^\ell(S) := \sum_{u \in (\mathbb{F}_2^n)^\ell} \chi_s(u) \cdot \mathbb{1}_{\mathrm{span}(u)=U}$$

where $s \in (\mathbb{F}_2^n)^\ell$ *is any* ℓ*-tuple with* $\mathrm{span}(s) = S$.

It is annoying that this function on subspaces depends on the parameter ℓ, but we do not know how to formulate a hierarchy that is more basis-independent.

Define the linear programming hierarchy $\mathrm{SubspaceLP}_{\mathrm{Lin}}(n, d, \ell)$ by:

$$\boxed{\begin{array}{llll}
 & \text{Variables: } a_S & S \in \mathrm{Gr}(\leq\ell, \mathbb{F}_2^n) & \\
\max & \displaystyle\sum_{S \in \mathrm{Gr}(\leq\ell, \mathbb{F}_2^n)} a_S & & \\
\text{s.t.} & a_0 = 1 & & \text{(Normalization)} \\
 & a_S = 0 & \exists x \in S.\ |x| \in \{1, \dots, d-1\} & \text{(Distance constraints)} \\
 & \displaystyle\sum_{S \in \mathrm{Gr}(\leq\ell, \mathbb{F}_2^n)} a_S K_U^\ell(S) \geq 0 & \forall U \in \mathrm{Gr}(\leq\ell, \mathbb{F}_2^n) & \text{(Fourier coefficients)} \\
 & a_S \geq 0 & \forall S \in \mathrm{Gr}(\leq\ell, \mathbb{F}_2^n) & \text{(Non-negativity).}
\end{array}}$$

Following the symmetrization as in the previous section, we have the following lemma.

Lemma 7.25. *For every* $n, \ell \in \mathbb{N}_+$ *and every* $d \in \{0, 1, \dots, n\}$, *we have*

$$\mathrm{val}(\mathrm{SubspaceLP}_{\mathrm{Lin}}(n, d, \ell)) = \mathrm{val}(\mathrm{KravchukLP}_{\mathrm{Lin}}(n, d, \ell))\,.$$

Suppose now that we symmetrize over the larger group action of $\mathrm{GL}_\ell(\mathbb{F}_2) \times S_n$, which gives another reasonable definition of the higher-order Kravchuk polynomials and linear program. There is one Kravchuk polynomial and one free variable for each orbit of this action.

Definition 7.26 (Fully symmetrized higher-order Kravchuks). *Let $O := (\mathbb{F}_2^n)^\ell / (\mathrm{GL}_\ell(\mathbb{F}_2) \times S_n)$ be the set of orbits of the $(\mathrm{GL}_\ell(\mathbb{F}_2) \times S_n)$-action as above. For each $h \in O$ we define the higher-order Kravchuk polynomial $K_h \colon O \to \mathbb{R}$ by*

$$K_h(g) := \sum_{(\alpha_1,\ldots,\alpha_\ell)\in h} \prod_{j=1}^{\ell} \chi_{\alpha_j}(x_j),$$

where $(x_1, \ldots, x_\ell)$ is any element in the orbit $g \in O$.

Since the symmetry group is larger and the number of orbits is smaller, the size of the resulting LP is minimized. However, since $|\mathrm{GL}_\ell(\mathbb{F}_2)| = \prod_{t=0}^{\ell-1}(2^\ell - 2^t) = O_\ell(1)$, for a constant ℓ, this only decreases the size of KravchukLP by a constant factor. For practical computations, constant factors make a difference and this symmetrization should likely be performed. For theoretical analysis, KravchukLP may be preferred because the orbits are simpler to describe (being captured by explicit combinatorial objects, configuration functions).

A combinatorial interpretation of $(\mathrm{GL}_\ell(\mathbb{F}_2) \times S_n)$-orbits as "subspace weight profiles" is as follows. The right action of S_n naturally induces an action over linear subspaces of $\mathbb{F}_2^n$ given by

$$W \cdot \sigma := \{w \cdot \sigma \mid w \in W\} \quad (W \leq \mathbb{F}_2^n, \sigma \in S_n).$$

It is straightforward to see that two ℓ-tuples $(x_1, \ldots, x_\ell), (y_1, \ldots, y_\ell) \in (\mathbb{F}_2^n)^\ell$ are in the same $(\mathrm{GL}_\ell(\mathbb{F}_2) \times S_n)$-orbit if and only if $\mathrm{span}\{x_1, \ldots, x_\ell\}$ and $\mathrm{span}\{y_1, \ldots, y_\ell\}$ are in the same S_n-orbit, which in turn is equivalent to saying that both spaces have the same dimension, say k, and there are ordered bases $b^x = (b^x_1, \ldots, b^x_k)$ and $b^y = (b^y_1, \ldots, b^y_k)$ of these spaces respectively such that $\mathsf{Config}^\Delta_{n,k}(b^x) = \mathsf{Config}^\Delta_{n,k}(b^y)$. Thus, the hierarchy corresponding to the $(\mathrm{GL}_\ell(\mathbb{F}_2) \times S_n)$-action has an interesting interpretation as measuring weight statistics of linear subspaces of the linear code of dimension at most ℓ.

7.3.4 The Hierarchy as an SDP

The LP hierarchy is also equivalent to an SDP relaxation with the harsh constraint that the SDP matrix must be *translation invariant*.

Define the semidefinite program TranslationSDP(n, d, ℓ) as

$$\boxed{\begin{array}{llll}
 & \text{Variable: } M \in \mathbb{R}^{(\mathbb{F}_2^n)^\ell \times (\mathbb{F}_2^n)^\ell} & \text{symmetric} & \\
\max & \displaystyle\sum_{x \in (\mathbb{F}_2^n)^\ell} M[0, x] & & \\
\text{s.t.} & M[0,0] = 1 & & \text{(Normalization)} \\
 & M[0,(x_1,\ldots,x_\ell)] = 0 & \exists i \in [\ell].\ |x_i| \in \{1,\ldots,d-1\} & \text{(Distance constraints)} \\
 & M[x,y] = M[0, y-x] & \forall x, y \in (\mathbb{F}_2^n)^\ell & \text{(Translation symmetry)} \\
 & M \succeq 0 & & \text{(PSD-ness)} \\
 & M[x,y] \geq 0 & \forall x, y \in (\mathbb{F}_2^n)^\ell & \text{(Non-negativity)}
\end{array}}$$

To form TranslationSDP$_{\text{Lin}}(n, d, \ell)$, replace the distance constraints by

$$M[0,(x_1,\ldots,x_\ell)] = 0 \quad \exists w \in \text{span}(x_1,\ldots,x_\ell), |w| \in \{1,\ldots,d-1\}.$$

The crucial translation symmetry property of TranslationSDP ensures M lies in the *commutative* matrix algebra span$\{D_z \mid z \in (\mathbb{F}_2^n)^\ell\}$, where

$$D_z[x,y] := \mathbb{1}[y - x = z].$$

The coefficient of M on D_z is $M[0,z]$.

Since the matrices D_z commute, they are simultaneously diagonalizable. More specifically, their common eigenvectors are the Fourier characters.

Fact 7.27. *The matrices D_z are simultaneously diagonalized by $(\chi_\alpha \mid \alpha \in (\mathbb{F}_2^n)^\ell)$ with the eigenvalue of D_z on χ_α being $\chi_\alpha(z)$.*

Therefore, the PSD-ness constraint in TranslationSDP is particularly simple: to check that $\lambda_z D_z \succeq 0$, it is equivalent to check $\sum_{z \in (\mathbb{F}_2^n)^\ell} \lambda_z \chi_\alpha(z) \geq 0$ for all $\alpha \in (\mathbb{F}_2^n)^\ell$. This is a linear constraint on the λ_z, and hence we can express the SDP as an LP, giving the FourierLP formulation of the hierarchy.

Proposition 7.28. *For every $n, \ell \in \mathbb{N}_+$ and every $d \in \{0, 1, \ldots, n\}$, we have*

$$\begin{aligned}
\text{val}(\text{FourierLP}(n,d,\ell)) &= \text{val}(\text{TranslationSDP}(n,d,\ell)), \\
\text{val}(\text{FourierLP}_{\text{Lin}}(n,d,\ell)) &= \text{val}(\text{TranslationSDP}_{\text{Lin}}(n,d,\ell)).
\end{aligned}$$

Proof. The formal correspondence of the variables is $M[0,x] = a_x$. The Fourier coefficient constraints in FourierLP are equivalent to PSD-ness as described above, and the other constraints also match up. □

Along with Proposition 7.20, the above implies that TranslationSDP also has the same value as KravchukLP.

Remark 7.29. *In previous convex relaxations for $A_2(n,d)$, in order to implement the program* *efficiently, a key technical step has been finding an explicit block diagonalization of the SDP matrix (which reduces the program size). This step requires significant technical work [Sch05, GMS12, Gij09]. An advantage of the LP hierarchy is that complete diagonalization is trivial.*

7.3.5 The Hierarchy as ϑ'

The hierarchy can also be seen as computing the (modified) Lovász ϑ' function on progressively larger graphs, whose definition is recalled below. In fact, this formulation of the hierarchy holds for any *association scheme*.

Definition 7.30 (ϑ' Program)**.** *The (modified) Lovász ϑ' function is defined as follows. For a graph G, $\vartheta'(G)$ is the optimum value of the semidefinite program $\mathcal{S}(G)$ given by:*

$$\boxed{\begin{array}{llll}
 & \text{Variables: } M \in \mathbb{R}^{V(G)\times V(G)} & \text{symmetric} & \\
\max & \langle J, M\rangle & & \\
\text{s.t.} & \operatorname{tr} M = 1 & & \text{(Normalization)} \\
 & M[u,v] = 0 & \forall \{u,v\} \in E(G) & \text{(Independent set)} \\
 & M \succeq 0 & & \text{(PSD-ness)} \\
 & M[u,v] \geq 0 & \forall u,v \in V(G) & \text{(Non-negativity)}
\end{array}}$$

where J is the all ones matrix and $\langle A, B\rangle := \operatorname{tr}(A^\top B)$.

By strong duality $\vartheta'(G)$ is also the optimum value of the dual semidefinite program $\mathcal{S}'(G)$ given by:

$$\boxed{\begin{array}{llll}
 & \text{Variables: } N \in \mathbb{R}^{V(G)\times V(G)} & \text{symmetric} & \\
 & \qquad\qquad\;\; \beta \in \mathbb{R} & & \\
\min & \beta & & \\
\text{s.t.} & \beta I - N \succeq 0 & & \text{(PSD-ness)} \\
 & N[u,v] \geq 1 & \forall u,v \in V(G) \text{ with } \{u,v\} \notin E & \text{(Independent set)}.
\end{array}}$$

It is straightforward to see that $\vartheta'(G)$ is an upper bound for the independence number of the graph G since if $A \subseteq V(G)$ is an independent set, then $\mathbb{1}_A \mathbb{1}_A^\top / |A|$ is a feasible solution of $\mathcal{S}(G)$ with value $|A|$.

In the same way that a code $C \subseteq \mathbb{F}_2^n$ of distance at least d can be seen as an independent set in the graph $H_{n,d}$, we can see C^ℓ as an independent set in exclusion graphs defined below based on the sets $\mathsf{ForbConfig}(n,d,\ell)$ and $\mathsf{ForbConfig}_{\mathrm{Lin}}(n,d,\ell)$ of Definition 7.19.

Definition 7.31 (Exclusion Graph)**.** *We define the exclusion graph $H_{n,d,\ell}$ to have vertex set*

$(\mathbb{F}_2^n)^\ell$ and edge set

$$E(H_{n,d,\ell}) := \left\{ (x,y) \in \binom{(\mathbb{F}_2^n)^\ell}{2} \;\middle|\; \mathsf{Config}^{\Delta}_{n,\ell}(x-y) \in \mathsf{ForbConfig}(n,d,\ell) \right\}.$$

We define $H^{\mathrm{Lin}}_{n,d,\ell}$ analogously replacing $\mathsf{ForbConfig}(n,d,\ell)$ with $\mathsf{ForbConfig}_{\mathrm{Lin}}(n,d,\ell)$.

Lemma 7.32. *For every $n, \ell \in \mathbb{N}_+$ and every $d \in \{0,1,\dots,n\}$, we have*

$$\begin{aligned} \mathrm{val}(\mathrm{TranslationSDP}(n,d,\ell)) &= \mathrm{val}(\vartheta'(H_{n,d,\ell})), \\ \mathrm{val}(\mathrm{TranslationSDP}_{\mathrm{Lin}}(n,d,\ell)) &= \mathrm{val}(\vartheta'(H^{\mathrm{Lin}}_{n,d,\ell})). \end{aligned}$$

Proof. The program $\mathcal{S}(H_{n,d,\ell})$ corresponding to $\vartheta'(H_{n,d,\ell})$ is invariant under $\mathrm{Aut}(G)$, so it is invariant in particular under the translation action of $\mathbb{F}_2^n$ on itself.

Therefore, by Fact 7.14, we may consider only solutions of $\mathcal{S}(H_{n,d,\ell})$ that are translation invariant. Now there is a correspondence between solutions M for TranslationSDP and translation invariant solutions M' for $\mathcal{S}(H_{n,d,\ell})$ given by $M = 2^{n\ell} \cdot M'$. The proof goes through similarly for the linear case. □

7.3.6 Comparison with the Sum-of-Squares Hierarchy

The sum-of-squares hierarchy can also be run out-of-the-box on the Independent Set formulation of $A_2(n,d)$. Here we prove that our hierarchy is weaker than the sum-of-squares hierarchy. In the case of linear codes, the sum-of-squares hierarchy needs to be augmented with additional semantic constraints. On the other hand, our hierarchy has the advantage that it is simpler than sum-of-squares yet still complete.

Because our hierarchy may be able to prove new upper bounds on $A_2(n,d)$, sum-of-squares reasoning may be able to prove new upper bounds. This suggests that current approaches to this problem using suitably powerful hypercontractive inequalities have a chance of succeeding [Sam21].

The sum-of-squares hierarchy is applied to the polynomial encoding of Independent Set where for a graph $G = (V(G), E(G))$, we take a set of variables $\mathsf{X} = \{\mathsf{X}_v\}_{v \in V(G)}$ and use the polynomial system,

$$\begin{aligned} &\max \sum_{v \in V(G)} \mathsf{X}_v \\ &\text{s.t.} \\ &\mathsf{X}_v^2 = \mathsf{X}_v \qquad \forall v \in V(G) \qquad \text{(Booleanity)} \\ &\mathsf{X}_u \mathsf{X}_v = 0 \qquad \forall \{u,v\} \in E(G) \qquad \text{(Independent Set)}\,. \end{aligned}$$

The sum-of-squares algorithm from Chapter 3 may be applied to this polynomial system, which results in the following SDP relaxation at level ℓ of the sum-of-squares hierarchy.

Definition 7.33 (Sum-of-squares relaxation $\mathrm{SoS}(G, 2\ell)$)**.** *The degree-2ℓ sum-of-squares relaxation for Independent set, denoted $\mathrm{SoS}(G, 2\ell)$, is equivalent to the following SDP. The variable is $M \in \mathbb{R}^{(\varnothing \cup V(G)^{\ell}) \times (\varnothing \cup V(G)^{\ell})}$ where the row/column index set is to be interpreted as "the empty set or $V(G)^{\ell}$".*

$$
\boxed{
\begin{array}{llll}
 & \text{Variables: } M \in \mathbb{R}^{(\varnothing \cup V(G)^{\ell}) \times (\varnothing \cup V(G)^{\ell})} & \text{symmetric} & \\
 & \max \sum\limits_{v \in V(G)} M[(v, \dots, v), (v, \dots v)] & & \\
\text{s.t.} & & & \\
 & M[\varnothing, \varnothing] = 1 & & \text{(Normalization)} \\
 & M[(u_1, \dots, u_\ell), (v_1, \dots, v_\ell)] = 0 & \text{if } \exists i, j \in [\ell]. \{u_i, v_j\} \in E(G) & \text{(Independent set)} \\
 & M[I, J] = M[I', J'] & \forall I \cup J = I' \cup J' \text{ as sets} & \begin{array}{c}\text{(SoS symmetry,} \\ \text{Booleanity)}\end{array} \\
 & M \succeq 0 & & \text{(PSD-ness)}
\end{array}
}
$$

For linear codes, we can add in the additional semantic constraints on distances to sum-of-squares. Let $\mathrm{SoS}_{\mathrm{Lin}}(H_{n,d}, \ell)$ be $\mathrm{SoS}(H_{n,d}, \ell)$ with the independent set constraints replaced by the "distance constraints"

$$M[(u_1, \dots, u_\ell), (v_1, \dots, v_\ell)] = 0 \quad \text{if } \exists x \in \mathrm{span}(u_1, \dots, u_\ell, v_1, \dots, v_\ell).\ |x| \in \{1, \dots, d-1\}.$$

The following proposition states that sum-of-squares is stronger than our hierarchy. The intuitive reason for this is that, for $(u_1, \dots, u_\ell), (v_1, \dots, v_\ell) \in (\mathbb{F}_2^n)^{\ell}$, the sum-of-squares matrix entry $M[(u_1, \dots, u_\ell), (v_1, \dots, v_\ell)]$ checks the distance constraint on the space $\mathrm{span}(u_1, \dots, v_1, \dots)$, whereas in our hierarchy the distance constraint is only on $\mathrm{span}(u_1 - v_1, \dots, u_\ell - v_\ell)$.

Proposition 7.34.

$$\mathrm{val}(\mathrm{SoS}(H_{n,d}, 4\ell)) \leq \mathrm{val}(\mathrm{KravchukLP}(n, d, \ell))^{1/\ell}$$
$$\mathrm{val}(\mathrm{SoS}_{\mathrm{Lin}}(H_{n,d}, 4\ell)) \leq \mathrm{val}(\mathrm{KravchukLP}_{\mathrm{Lin}}(n, d, \ell))^{1/\ell}$$

Proof. We will interchangeably use the interpretation of an SoS solution as a moment matrix $M_{4\ell}$ or a pseudoexpectation operator $\widetilde{\mathbb{E}}$.

We compare SoS with the $\vartheta'(H_{n,d,\ell})$ formulation of the hierarchy, Lemma 7.32. Let $M_{4\ell} \in \mathbb{R}^{(\varnothing \cup (\mathbb{F}_2^n)^{2\ell}) \times (\varnothing \cup (\mathbb{F}_2^n)^{2\ell})}$ be a feasible SoS solution of degree 4ℓ. It suffices to construct a feasible solution to $\vartheta'(H_{n,d,\ell})$ with the same value. The candidate solution $M \in \mathbb{R}^{(\mathbb{F}_2^n)^{\ell} \times (\mathbb{F}_2^n)^{\ell}}$ is formed by taking

$$M[x, y] = M_{4\ell}[(x, x), (y, y)] = \widetilde{\mathbb{E}}\left[\prod_{i=1}^{\ell} \mathsf{X}_{x_i} \mathsf{X}_{y_i}\right]$$

and then normalizing the matrix by $\mathrm{tr}(M)$.

Normalization and the independent set constraints follow definitionally. PSD-ness follows from the fact that M is a principle submatrix of $M_{4\ell}$. For positivity, PSD-ness of $M_{4\ell}$ implies that all diagonal entries of $M_{4\ell}$ are positive. By SoS symmetry and the fact that $x \cup y$ contains at most 2ℓ strings, every entry of M appears somewhere on the diagonal of $M_{4\ell}$. Therefore every entry of M is non-negative, and M is feasible.[3]

The objective value of M is:

$$\begin{aligned}\langle J, M\rangle &= \frac{\sum_{x,y \in (\mathbb{F}_2^n)^\ell} \widetilde{\mathbb{E}}\left[\prod_{i=1}^{\ell} \mathsf{X}_{x_i}\mathsf{X}_{y_i}\right]}{\sum_{x \in (\mathbb{F}_2^n)^\ell} \widetilde{\mathbb{E}}\left[\prod_{i=1}^{\ell} \mathsf{X}_{x_i}\right]} \\ &= \frac{\widetilde{\mathbb{E}}\left[\left(\sum_{x \in \mathbb{F}_2^n} \mathsf{X}_x\right)^{2\ell}\right]}{\widetilde{\mathbb{E}}\left[\left(\sum_{x \in \mathbb{F}_2^n} \mathsf{X}_x\right)^{\ell}\right]}.\end{aligned}$$

We want to claim this is at least $\mathrm{val}(\mathrm{SoS}(H_{n,d}, 4\ell))^\ell = \widetilde{\mathbb{E}}\left[\sum_{x \in \mathbb{F}_2^n} \mathsf{X}_x\right]^\ell$. If $\widetilde{\mathbb{E}}$ were a real expectation operator, then two applications of a basic fact about monotonicity of moments would prove it.

Fact 7.35. *For $k \in \mathbb{N}$, if Z is a (sufficiently integrable) random variable such that $Z \geq 0$ almost surely, then $\mathbb{E}[Z^k] \geq \mathbb{E}[Z]^k$.*

Since this fact with $k = 2\ell$ has a degree-2ℓ SoS proof (Fact 3.20), the same conclusion holds when using the pseudoexpectation operator. □

7.4 Main Properties of the Kravchuk Hierarchies

This section presents our main results on the linear programming hierarchy.

The first result is the completeness of the higher-order linear programming hierarchies for *linear* codes. We show that for linear codes the hierarchy $\mathrm{KravchukLP}_{\mathrm{Lin}}(n, d, \ell)$ is *complete* at level $\ell = n$, meaning that it recovers the correct value $\mathrm{val}(\mathrm{KravchukLP}_{\mathrm{Lin}}(n, d, n))^{1/n} = A_2^{\mathrm{Lin}}(n, d)$.

The second result is the collapse of the hierarchies for *general* codes. Not unexpectedly, the hierarchy $\mathrm{KravchukLP}(n, d, \ell)$ corresponding to general (not necessarily linear) codes does not improve on Delsarte's linear program. Without the extra structure of linearity, the number of constraints we can add to our LP hierarchy is limited. We prove that solutions of $\mathrm{DelsarteLP}(n, d)$ (easily) lift to solutions of $\mathrm{KravchukLP}(n, d, \ell)$ with the same value.

[3] Except for the positivity constraint, SoS degree 2ℓ is sufficient to match level ℓ of KravchukLP.

This contrast between the hierarchies $\mathrm{KravchukLP}_{\mathrm{Lin}}(n,d,\ell)$ and $\mathrm{KravchukLP}(n,d,\ell)$ reinvigorates the question of whether the maximum sizes of general and linear codes are substantially different or not.

7.4.1 Completeness for Linear Codes

We prove that:

Theorem 7.36. *For $n,d \in \mathbb{N}^+$, $\mathrm{val}(\mathrm{KravchukLP}(n,d,n)) = A_2^{\mathrm{Lin}}(n,d)^n$.*

Intuitively, this is true because at level n, the feasible region of the LP already encodes $A_2^{\mathrm{Lin}}(n,d)$. That is, since at level n there is a variable for each possible basis of a subspace of $\mathbb{F}_q^n$, just writing down the distance constraints of $\mathrm{KravchukLP}_{\mathrm{Lin}}(n,d,n)$ allows one to deduce the true value of $A_2^{\mathrm{Lin}}(n,d)$. However, the LP hierarchy does not know how to use that kind of reasoning, and hence our proof is more involved.

Note that we *do not* show that the polytope is integral, meaning that all vertices of the polytope are true codes (and therefore all feasible points are convex combinations of true codes). This is a stronger condition than completeness, but we believe it is not true for $\mathrm{KravchukLP}(n,d,n)$ as we have defined it.

It is plausible that exact completeness of $\mathrm{KravchukLP}_{\mathrm{Lin}}(n,d,\ell)$ can be attained at level $O(k_0)$, where k_0 is the dimension of an optimum linear code over $\mathbb{F}_2$ of distance d and blocklength n.

Proof of Theorem 7.36. Recall the unsymmetrized formulation of the hierarchy from Section 7.3.1.

$$\boxed{\begin{array}{llll}
 & \text{Variables: } a_x & x \in (\mathbb{F}_2^n)^\ell & \\
\max & \displaystyle\sum_{x \in (\mathbb{F}_2^n)^\ell} a_x & & \\
\text{s.t.} & a_0 = 1 & & \text{(Normalization)} \\
 & a_{(x_1,\ldots,x_\ell)} = 0 & \exists w \in \mathrm{span}(x_1,\ldots,x_\ell).\ |w| \in [d-1]\} & \text{(Distance constraints)} \\
 & \displaystyle\sum_{x \in (\mathbb{F}_2^n)^\ell} a_x \chi_\alpha(x) \geq 0 & \forall \alpha \in (\mathbb{F}_2^n)^\ell & \text{(Fourier coefficients)} \\
 & a_x \geq 0 & \forall x \in (\mathbb{F}_2^n)^\ell & \text{(Non-negativity).}
\end{array}}$$

Let $k_0 = \log_2 A_2^{\mathrm{Lin}}(n,d)$ be the maximum dimension of a distance-d linear code. We may weaken the distance constraints to the following "dimension constraints". This is a weakening by the definition of k_0.

$$\boxed{a_{(x_1,\ldots,x_\ell)} = 0 \qquad \text{if } \dim(\mathrm{span}(x_1,\ldots,x_\ell)) > k_0 \qquad \text{(Dimension constraints)}}$$

It suffices to show that this weakened program still has value $A_2^{\mathrm{Lin}}(n,d)^\ell$ (which is still intuitively true, since writing down the dimension constraints is enough to determine k_0).

By symmetrizing over the basis change action of $\mathrm{GL}_\ell(\mathbb{F}_2)$ as in Section 7.3.3, we may assume that $a_x = a_y$ if $\mathrm{span}(x) = \mathrm{span}(y)$. We will use S instead of x to keep in mind that $S \leq \mathbb{F}_2^n$ is a subspace of $\mathbb{F}_2^n$.

Each of the variables a_S is the relaxation of the indicator $\mathbb{1}_{S \subseteq C}$ for a code C. For intuition, we rename the variable as $a_S = \widetilde{\Pr}[S \subseteq \mathsf{C}]$ where C is a formal variable that represents a code drawn from the pseudodistribution.

The polytope is integral if and only if the pseudodistribution is a true distribution on codes, viz. $\widetilde{\Pr}[S = \mathsf{C}]$ form a valid probability distribution. Formally, these are formal variables which are a wisely-chosen linear transformation of the variables $\widetilde{\Pr}[S \subseteq \mathsf{C}]$, defined by the system of linear equations:

$$\widetilde{\Pr}[S \subseteq \mathsf{C}] = \sum_{T \geq S} \widetilde{\Pr}[T = \mathsf{C}] \qquad (S \leq \mathbb{F}_2^n)\,.$$

The idea of the proof is to rewrite the linear program in terms of the variables $\widetilde{\Pr}[S = \mathsf{C}]$ and then argue about the program from the perspective of the pseudoprobabilities. The notation is shortened to $\widetilde{\Pr}[S] = \widetilde{\Pr}[S = \mathsf{C}]$.

- For $\ell \geq n$, the rewritten objective function is

$$\begin{aligned}\sum_{x \in (\mathbb{F}_2^n)^\ell} \widetilde{\Pr}[\mathrm{span}(x) \subseteq \mathsf{C}] &= \sum_{x \in (\mathbb{F}_2^n)^\ell} \sum_{T \geq \mathrm{span}(x)} \widetilde{\Pr}[T] \\ &= \sum_{S \leq \mathbb{F}_2^n} |S|^\ell \, \widetilde{\Pr}[S]\,.\end{aligned}$$

- The dimension constraints are

$$\sum_{T \geq S} \widetilde{\Pr}[T] = 0 \qquad (S \leq \mathbb{F}_2^n : \ \dim(S) > k_0)\,.$$

  By induction downwards on the dimension of S, equivalently

$$\widetilde{\Pr}[S] = 0 \qquad (S \leq \mathbb{F}_2^n : \ \dim(S) > k_0)$$

- The left-hand side of the Fourier constraint for α is

$$\begin{aligned}\sum_{x \in (\mathbb{F}_2^n)^\ell} \widetilde{\Pr}[\mathrm{span}(x) \subseteq \mathsf{C}] \chi_\alpha(x) &= \sum_{x \in (\mathbb{F}_2^n)^\ell} \chi_\alpha(x) \sum_{T \geq \mathrm{span}(x)} \widetilde{\Pr}[T] \\ &= \sum_{S \leq \mathbb{F}_2^n} \widetilde{\Pr}[S] \sum_{x \in S^\ell} \chi_\alpha(x)\,.\end{aligned}$$

  The inner summation often cancels to zero, as shown by the following lemma.

Lemma 7.37. *For $\alpha \in (\mathbb{F}_2^n)^\ell, S \leq \mathbb{F}_2^n$,*

$$\sum_{x \in S^\ell} \chi_\alpha(x) = \begin{cases} |S|^\ell & S \leq \operatorname{span}(\alpha)^\perp \\ 0 & \textit{otherwise} \end{cases} .$$

Proof. If $S \subseteq \operatorname{span}(\alpha)^\perp$, then $\chi_{\alpha_i}(x_i) = (-1)^{\langle \alpha_i, x_i \rangle} = 1$ for all α_i and $x_i \in S$. The summation is $|S|^\ell$. Conversely, if S contains some vector s such that $\left\langle s, \sum_{i=1}^\ell b_i \alpha_i \right\rangle = 1$, for some coefficients $b_i \in \mathbb{F}_2$, then there is a cancellation between terms

$$(x_1, \ldots, x_\ell) \in S^\ell \qquad \text{and} \qquad (x_1 + b_1 s, \ldots, x_\ell + b_\ell s) \in S^\ell .$$

The b_i cannot all be 0, hence the terms in the sum pair up and cancel. □

Therefore the Fourier constraint for α is

$$\sum_{S \leq \operatorname{span}(\alpha)^\perp} |S|^\ell \, \widetilde{\Pr}[S] \geq 0 .$$

In summary, KravchukLP(n, d, ℓ) (when weakened to the dimension constraints) is formulated in terms of pseudoprobabilities as the following linear program.

	Variables: $\widetilde{\Pr}[S]$	$S \leq \mathbb{F}_2^n$	
max	$\sum_{S \leq \mathbb{F}_2^n} \lvert S\rvert^\ell \, \widetilde{\Pr}[S]$		
s.t.	$\sum_{S \leq \mathbb{F}_2^n} \widetilde{\Pr}[S] = 1$		(Normalization)
	$\widetilde{\Pr}[S] = 0$	if $\dim(S) > k_0$	(Dimension constraints)
	$\sum_{S \leq U} \lvert S\rvert^\ell \, \widetilde{\Pr}[S] \geq 0$	$\forall U \leq \mathbb{F}_2^n$	(Fourier coefficients)
	$\sum_{T \geq S} \widetilde{\Pr}[T] \geq 0$	$\forall S \leq \mathbb{F}_2^n$	(Non-negativity).

We claim that optimal solutions of this program are integral, i.e. they have $\widetilde{\Pr}[S] \geq 0$ for all $S \leq \mathbb{F}_2^n$. This is sufficient to finish the proof, since integral points are a true probability distribution on subspaces S of dimension at most k_0, and hence the value is upper bounded by the true value $2^{k_0 \ell} = A_2^{\mathrm{Lin}}(n, d)^\ell$.

Let $\widetilde{\Pr}$ be an optimal point. Let $S_{\min}$ be a subspace of minimum dimension in the support of $\widetilde{\Pr}$. By the Fourier constraint on $S_{\min}$, $\widetilde{\Pr}[S_{\min}] \geq 0$ for any minimal subspace in the support of $\widetilde{\Pr}$. We show that $\dim(S_{\min}) = k_0$, i.e. that $\widetilde{\Pr}$ is supported on spaces of dimension exactly k_0, to conclude that $\widetilde{\Pr}$ is nonnegative.

To show that $\dim(S_{\min}) = k_0$, assume for the sake of contradiction that $\dim(S_{\min}) < k_0$. Then there is a way to increase the value of $\widetilde{\Pr}$, a contradiction. Indeed, we may divide the probability mass equally among the spaces $S \geq S_{\text{min}}$ with $\dim(S) = \dim(S_{\text{min}}) + 1$. Letting m be the number of such spaces, formally we define:

$$\widetilde{\Pr}_+[S] = \begin{cases} 0 & S = S_{\text{min}} \\ \widetilde{\Pr}[S] + \frac{\widetilde{\Pr}[S_{\text{min}}]}{m} & S \geq S_{\text{min}} \text{ and } \dim(S) = \dim(S_{\text{min}}) + 1 \\ \widetilde{\Pr}[S] & \text{otherwise} \end{cases} \cdot$$

This evidently increases the objective function. Let us verify that $\widetilde{\Pr}_+$ remains a feasible solution.

- $\widetilde{\Pr}_+$ respects the normalization $\sum_{S \leq \mathbb{F}_2^n} \widetilde{\Pr}_+[S] = 1$.
- The dimension constraints are not violated since $\dim(S) = \dim(S_{\text{min}}) + 1 \leq k_0$.
- In Fourier constraints with $U \ngeq S_{\text{min}}$, nothing changes. In $U = S_{\text{min}}$, the left-hand side is 0. In $U > S_{\text{min}}$, U contains at least one of the subspaces S with increased mass. Therefore the change in the left-hand side is at least

  $$|S|^{\ell} \cdot \frac{\widetilde{\Pr}[S_{\text{min}}]}{m} - |S_{\text{min}}|^{\ell} \cdot \widetilde{\Pr}[S_{\text{min}}] \,.$$

  Since $m \leq 2^n$ while $\frac{|S|^{\ell}}{|S_{\text{min}}|^{\ell}} = 2^{\ell} \geq 2^n$, this is nonnegative.
- The non-negativity constraints remain satisfied.

Therefore, $\widetilde{\Pr}$ must be supported only on spaces of dimension exactly k_0, it is nonnegative and integral, and the proof is complete. □

7.4.2 Hierarchy Collapse for General Codes

In this section, we show that without the additional semantic linearity constraints imposed by $\text{KravchukLP}_{\text{Lin}}(n, d, \ell)$, the associated hierarchy $\text{KravchukLP}(n, d, \ell)$ does not give any improvement over the original Delsarte linear programming approach.

Theorem 7.38 (Lifting). *For every $n, d, \ell \in \mathbb{N}_+$,*

$$\text{val}(\text{KravchukLP}(n, d, \ell))^{1/\ell} = \text{val}(\text{DelsarteLP}(n, d)).$$

Proof. We will use the ϑ' formulation of the program from Lemma 7.32. We have

$$\text{val}(\text{KravchukLP}(n, d, \ell)) = \vartheta'(H_{n,d,\ell}) \,.$$

The graph $H_{n,d,\ell}$ is just the tensor product $H_{n,d}^{\otimes \ell}$, reflecting the fact that KravchukLP only enforces a constraint on $(x_1, \ldots, x_\ell), (y_1, \ldots, y_\ell)$ by looking separately at each index $i \in$

$[\ell]$,

$$|x_i - y_i| \notin \{1, \ldots, d-1\}\,.$$

It is easy to see that $\vartheta'(G^{\otimes \ell}) = \vartheta'(G)^\ell$ by tensoring the primal and dual solutions. Meanwhile, we have $\mathrm{val}(\mathrm{DelsarteLP}(n,d)) = \mathrm{val}(\mathrm{KravchukLP}(n,d,1)) = \vartheta'(H_{n,d})$, and therefore

$$\mathrm{val}(\mathrm{KravchukLP}(n,d,\ell)) = \vartheta'(H_{n,d,\ell}) = \vartheta'(H_{n,d})^\ell = \mathrm{val}(\mathrm{DelsarteLP}(n,d))^\ell\,.$$

□

7.5 Dual Program

For linear codes, we would like to use the LP hierarchy to improve upper bounds on $A_2^{\mathrm{Lin}}(n,d)$ beyond the Delsarte LP. The natural way to do this is to construct dual solutions to a higher level of the hierarchy. Let $\mathrm{FourierLP}'_{\mathrm{Lin}}(n,d,\ell)$ denote the dual linear program. For any level ℓ of the hierarchy, if we construct a feasible point f for the dual, then by weak duality we immediately prove:

$$A_2^{\mathrm{Lin}}(n,d) \le \mathrm{val}(f)^{1/\ell}.$$

As evidenced by the substantial effort invested in Chapter 4 and Chapter 6, constructing dual solutions for convex programs can be difficult.

We already developed one formulation of the dual program through the Lovász ϑ' function, Definition 7.30. Here we develop another formulation that closely parallels the dual of the Delsarte LP, for which good solutions were constructed by McEliece et al [MRRW77] (in the Boolean setting) and by Cohn and Elkies [CE03] and Viazovska [Via17] (in the sphere packing setting). The construction of solutions for this program is ongoing work.

In this section, we introduce the following notation.

Definition 7.39 (Valid)**.** *We say that $C \subseteq \mathbb{F}_2^n$ is valid if for all $w \in C$, $|w| = 0$ or $|w| \ge d$. If there is a word $w \in C$ with $|w| \in \{1, \ldots, d-1\}$, we say C is invalid. We also use the same terminology for $x = (x_1, \ldots, x_\ell) \in (\mathbb{F}_2^n)^\ell$ with reference to the subspace* $\mathrm{span}(x_1, \ldots, x_\ell)$.

We construct the dual from the $\mathrm{FourierLP}_{\mathrm{Lin}}(n,d,\ell)$ formulation of the hierarchy. Define the dual program $\mathrm{FourierLP}'_{\mathrm{Lin}}(n,d,\ell)$ as follows.

Variables:	$f : (\mathbb{F}_2^n)^\ell \to \mathbb{R}$	
min $f(0)$		
s.t. $\widehat{f}(0) = 1$		(Normalization)
$f(x) \le 0$	$\forall$ valid $x, x \ne 0$	(Distance constraints)
$\widehat{f}(\alpha) \ge 0$	$\forall \alpha \in (\mathbb{F}_2^n)^\ell$	(Fourier coefficients)

We can prove weak duality of these programs via a direct argument. (Strong duality we will show afterwards.)

Proposition 7.40. *For any pair of feasible solutions* a_x $(x \in (\mathbb{F}_2^n)^\ell)$ *to* $\mathrm{FourierLP}_{\mathrm{Lin}}(n,d,\ell)$ *and* $f : (\mathbb{F}_2^n)^\ell \to \mathbb{R}$ *to* $\mathrm{FourierLP}'_{\mathrm{Lin}}(n,d,\ell)$*,*

$$\mathrm{val}(a_x) \leq \mathrm{val}(f)\,.$$

Proof. Change a_x into a function $a' : (\mathbb{F}_2^n)^\ell \to \mathbb{R}$ by $a'(x) = 2^{-n\ell} a_x$.

$$\begin{aligned}
\mathrm{val}(a_x) &= \sum_{x \in (\mathbb{F}_2^n)^\ell} a_x \\
&= 2^{n\ell} \mathop{\mathbb{E}}_{x \in_{\mathrm{R}} (\mathbb{F}_2^n)^\ell} [a'(x)] \\
&= 2^{n\ell} \widehat{a'}(0) \widehat{f}(0) && \text{(Normalization of } f\text{)} \\
&\leq 2^{n\ell} \sum_{\alpha \in (\mathbb{F}_2^n)^\ell} \widehat{a'}(\alpha) \widehat{f}(\alpha) && \text{(Fourier coefficient constraints)} \\
&= 2^{n\ell} \mathop{\mathbb{E}}_{x \in (\mathbb{F}_2^n)^\ell} a'(x) f(x) && \text{(Plancherel identity, Fact 1.8)} \\
&\leq 2^{n\ell} a'(0) f(0) && \text{(Distance constraints and non-negativity)} \\
&= a(0) f(0) = f(0) && \text{(Normalization of } a\text{)} \\
&= \mathrm{val}(f)
\end{aligned}$$

On the other hand, $\widehat{a}(0) = \mathrm{val}(\mathrm{primal})$, finishing the claim. □

To prove strong duality, we show that this dual program is equivalent to the dual of the ϑ' formulation of the hierarchy, Lemma 7.32.

Lemma 7.41. $\mathrm{val}(\mathrm{FourierLP}'_{\mathrm{Lin}}(n,d,\ell)) \leq \vartheta'(H^{\mathrm{Lin}}_{n,d,\ell})$*. Along with the previous proposition, this implies* $\mathrm{val}(\mathrm{FourierLP}'_{\mathrm{Lin}}(n,d,\ell)) = \mathrm{val}(\mathrm{FourierLP}_{\mathrm{Lin}}(n,d,\ell))$*.*

Proof. We start by converting the dual $\vartheta'(H^{\mathrm{Lin}}_{n,d,\ell})$ SDP into an LP, as in Section 7.3. The program is invariant under the translation action of $\mathbb{F}_2^n$ on itself, therefore we may assume that the solution matrix is translation-invariant. The eigenvectors are therefore given by the Boolean Fourier characters. The PSD-ness constraint becomes a set of linear constraints, resulting in the following program.

$$\begin{array}{lll}
\text{Variables:} & c_z \quad (z \in (\mathbb{F}_2^n)^\ell) & \\
 & \beta \in \mathbb{R} & \\
\min \beta & & \\
\text{ s.t. } c_z \geq 1 & \forall \text{ valid } z & \text{(Distance constraints)} \\
\displaystyle\sum_{z \in (\mathbb{F}_2^n)^\ell} c_z \chi_\alpha(z) \leq \beta & \forall \alpha \in (\mathbb{F}_2^n)^\ell & \text{(Fourier coefficients)}
\end{array}$$

Let (c_z, β) be an optimal solution to this program. We may assume that $c_0 = 1$, since lowering c_0 can only aid the Fourier coefficient constraints. We define the candidate solution $f' : (\mathbb{F}_2^n)^\ell \to \mathbb{R}$ by:

$$f(x) = \begin{cases} 1 - c_x & x \neq 0 \\ \beta & x = 0 \end{cases}$$

and then normalizing $f' = \frac{f}{\widehat{f}(0)}$.

The distance constraints on f' are evidently satisfied. For the Fourier coefficient constraints, the Fourier coefficients of f are

$$\begin{aligned}
\widehat{f}(\alpha) &= \mathop{\mathbb{E}}_{x \in_{\mathrm{R}} (\mathbb{F}_2^n)^\ell} [f(x)\chi_\alpha(x)] \\
&= 2^{-n\ell}\beta + \mathop{\mathbb{E}}_{x \in_{\mathrm{R}} (\mathbb{F}_2^n)^\ell} [(1 - c_x)\chi_\alpha(x)] - 2^{-n\ell}(1 - c_0) \\
&= 2^{-n\ell}\beta + \mathop{\mathbb{E}}_{x \in_{\mathrm{R}} (\mathbb{F}_2^n)^\ell} [(1 - c_x)\chi_\alpha(x)] \\
&= \begin{cases} 2^{-n\ell}\beta - \widehat{c}(\alpha) & \alpha \neq 0 \\ 2^{-n\ell}\beta + 1 - \widehat{c}(0) & \alpha = 0 \end{cases}.
\end{aligned}$$

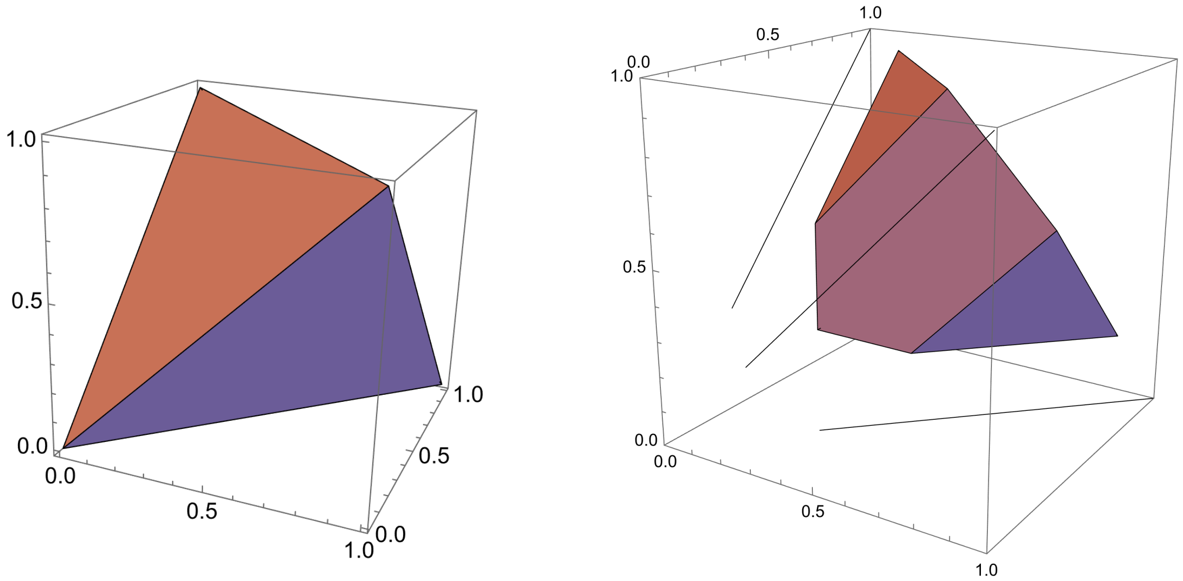
These are nonnegative via the Fourier constraints on c. This proves that f' is feasible. Moreover, the last equation above also shows $\widehat{f}(0) \geq 1$. The objective value is

$$f'(0) = \frac{\beta}{\widehat{f}(0)} \leq \beta\,.$$

Thus f shows $\mathrm{val}(\mathrm{FourierLP}'_{\mathrm{Lin}}(n, d, \ell)) \leq \vartheta(H^{\mathrm{Lin}}_{n,d,\ell})$ as desired.

For the interested reader, the explicit transformation in reverse sets $c_0 = 1$ and $c_x = 1 - f(x)$ for $x \neq 0$, and then $\beta = 2^{n\ell} \max_{\alpha \in (\mathbb{F}_2^n)^\ell} \widehat{c}(\alpha) = 2^{n\ell}\widehat{c}(0) = f(0)$. □

We work towards understanding the dual program. By symmetrizing, we may assume that a feasible function $f(x_1, \ldots, x_\ell)$ depends only on $\mathsf{Config}(x_1, \ldots, x_\ell)$. Recall that



(a) Locus of possible configurations for $\ell = 2$. The point (x, y, z) represents $\left(\frac{|x_1|}{n}, \frac{|x_2|}{n}, \frac{|x_1+x_2|}{n}\right)$ where $x_1, x_2 \in \mathbb{F}_2^n$.

(b) Valid configurations for which the distance constraints require $f(x) \leq 0$. The set consists of three lines and a polyhedral region. The polyhedral region is the intersection of the left picture with the cube $[d, 1]^3$.

Figure 7.1

a configuration specifies the Hamming weight of each linear combination of $x_1, \ldots, x_\ell$. Ignoring the trivial linear combination, which always has Hamming weight 0, the space of configurations can be visualized as a subset of $\{0, 1, \ldots, n\}^{2^\ell - 1}$. Not every element of this set arises from a configuration; the bona fide configurations lie in the polytope formed by 2^ℓ affine planes (essentially coming from the triangle inequality on Hamming weights). For example, the case $\ell = 2$ is visualized in Fig. 7.1a.

The distance constraints require that f is nonpositive on any valid configuration (except 0). The set of valid configurations can be concisely described as the set

$$(\{0\} \cup [d, n])^{2^\ell - 1} .$$

The case $\ell = 2$ is visualized in Fig. 7.1b.

One of the difficult aspects of studying the $\text{FourierLP}'_{\text{Lin}}(n, d, \ell)$ program is that the distance constraints are not on a connected subset. The subcube $[d, n]^{2^\ell - 1}$ consists of "most" of the valid configurations, but there are additional lower-dimensional spaces where one or more of the coordinates equal zero.

Since we are studying the regime where $\frac{d}{n} = \delta$ is a constant, the asymptotic rate $\lim_{n\to\infty} \frac{1}{n} \log_2 A_2^{\text{Lin}}(n, d)$ is a constant depending on δ but independent of n. Therefore, we would like to scale our hierarchy by $\frac{1}{n}$, take the limit $n \to \infty$, and characterize the asymptotic rate given by the hierarchy $\lim_{n\to\infty} \frac{1}{n} \log_2 \mathrm{val}(\mathrm{FourierLP}(n, d, \ell))$ with an expression that is independent of n.[4] The constraints in the two figures above behave nicely under this limit.

However, we do not know how to enforce the Fourier constraints under this limit. The distribution underlying the Fourier transform is the uniform distribution on $(\mathbb{F}_2^n)^\ell$. Let c be a random variable for the induced point in configuration space $\{0, 1, \dots, n\}^{2^\ell - 1}$. Given a uniform random sample $(x_1, \dots, x_\ell) \in_{\mathrm{R}} (\mathbb{F}_2^n)^\ell$, the entries of c are

$$c_S = \left| \sum_{i \in S} x_i \right| \qquad (S \subseteq [\ell], S \neq \emptyset)\,.$$

The distribution of c in the limit $n \to \infty$ can be understood. Each coordinate of c is individually a balanced binomial distribution, and the limiting version of c turns out to be a standard Gaussian centered around the point $(\frac{n}{2}, \dots, \frac{n}{2})$ with standard deviation on the order $\sqrt{n}$, exactly as if one took the limit of each coordinate independently.

Proposition 7.42.

$$\frac{c - (n/2, \dots, n/2)}{\sqrt{n}/2} \xrightarrow{d} \mathcal{N}(0, \mathrm{Id}_{2^\ell - 1})\,.$$

Proof. We can express c as the sum of n i.i.d. vectors v_i, one for each coordinate $i \in [n]$. The distribution of a single vector v is uniform over 2^ℓ choices. The choices are the columns of the standard Hadamard matrix, also known as the Hadamard code (with the tweak that the first row is removed, as it is all zeros).

Concretely, for each $\alpha \subseteq [\ell]$, the α-th choice is the parity of all nonempty subsets of α's coordinates. For example, for $\ell = 2$, the four vectors are:

$$\begin{array}{ccccc}
\text{Subsets} & \alpha = 00 & \alpha = 01 & \alpha = 10 & \alpha = 11 \\
\begin{bmatrix} \{1\} \\ \{2\} \\ \{1,2\} \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}
\end{array}$$

For $\ell = 3$, the eight vectors are:

$$\begin{array}{ccccccccc}
\text{Subsets} & \alpha = 000 & \alpha = 001 & \alpha = 010 & \alpha = 100 & \alpha = 011 & \alpha = 101 & \alpha = 110 & \alpha = 111 \\
\begin{bmatrix} \{1\} \\ \{2\} \\ \{3\} \\ \{1,2\} \\ \{1,3\} \\ \{2,3\} \\ \{1,2,3\} \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
\end{array}$$

[4] This technique is reminiscent of calculating the free energy density in statistical physics of n-particle systems [MM09].

The Gaussian limit of c is a consequence of the central limit theorem for sums of independent vectors,

$$\frac{c - n\,\mathbb{E}[v]}{\sqrt{n}} \xrightarrow{d} \mathcal{N}(0, \mathbb{E}[(v - \mathbb{E}[v])(v - \mathbb{E}[v])^{\intercal}])\,.$$

The mean of v is the all-1/2 point, while the covariance matrix of v is pleasantly just $\frac{1}{4}\mathrm{Id}$. Indeed, the uniform distribution on the Hadamard code is a canonical distribution such that the bits are uncorrelated but not independent (the distribution has only ℓ bits of entropy, yet the number of uncorrelated mean-zero bits is $2^{\ell} - 1$). To compute the correlation, for any $S, T \subseteq [\ell]$,

$$\mathop{\mathbb{E}}_{\alpha \subseteq [\ell]}[(v_S - 1/2)(v_T - 1/2)] = \frac{1}{4} \mathop{\mathbb{E}}_{\alpha \in_{\mathrm{R}} \{-1,+1\}^{\ell}}[\chi_S(\alpha)\chi_T(\alpha)] = \frac{1}{4}\mathbb{1}_{S=T}\,.$$

On the other hand, it is easy to see that the bits are not 3-wise independent. For any S, T, U such that $S \triangle T \triangle U = \emptyset$,

$$\mathop{\mathbb{E}}_{\alpha \subseteq [\ell]}[(v_S - 1/2)(v_T - 1/2)(v_U - 1/2)] = \frac{1}{8} \mathop{\mathbb{E}}_{\alpha \in_{\mathrm{R}} \{-1,+1\}^{\ell}}[\chi_S(\alpha)\chi_T(\alpha)\chi_U(\alpha)] = \frac{1}{8} \neq 0\,.$$

□

This is a nice distribution as $n \to \infty$, but it does not behave well under the $\frac{1}{n}$ rescaling. This distribution is highly concentrated around the all-$\frac{n}{2}$ point, and when rescaled it becomes a point mass in the limit $n \to \infty$.

7.6 Open Problems

The main open problem is to prove bounds on the value of the LP hierarchy for linear codes. Can analyzing the hierarchy for $\ell = 2$ prove a new upper bound on the maximum rate of a binary code?

Question 7.43. *For $\delta = \frac{d}{n} = \frac{1}{2} - \varepsilon$, what upper bound on the rate is proven by the hierarchy, $\lim_{n\to\infty} \frac{1}{\ell n} \log_2 \mathrm{KravchukLP}_{\mathrm{Lin}}(n, d, \ell)$? Is it strictly less than $\lim_{n\to\infty} \frac{1}{n} \log_2 \mathrm{DelsarteLP}(n, d)$?*

In Section 7.3.6, we showed that degree-4ℓ sum-of-squares is stronger than level-ℓ of the hierarchy. Both hierarchies are complete and have value $A_2^{\mathrm{Lin}}(n, d)$ at level $\ell = O(n)$. Do the hierarchies converge to $A_2^{\mathrm{Lin}}(n, d)$ at a similar rate, or is sum-of-squares faster?

Question 7.44. *Does $\mathrm{KravchukLP}_{\mathrm{Lin}}$ interleave the SoS hierarchy, in the sense that for a constant C and all $\ell \in \mathbb{N}$,*

$$\mathrm{val}(\mathrm{KravchukLP}_{\mathrm{Lin}}(n, d, C \cdot \ell))^{1/C\ell} \leq \mathrm{val}(\mathrm{SoS}_{\mathrm{Lin}}(H_{n,d}, \ell))\,?$$

Our techniques provide a higher-order version of the linear program responsible for the first linear programming bound in [MRRW77]. The second linear programming bound in [MRRW77] also consists of analyzing a Delsarte LP but for the Johnson scheme instead of the Hamming scheme. In the paper version of this chapter [CJJ22], we show an analogue of the hierarchy for codes in any translation scheme. However, since the Johnson scheme is not a translation scheme, one cannot apply the theory. It is then natural to ask if there is a suitable generalization of this construction that would apply to non-translation schemes such as the Johnson scheme.

Sphere packing exhibits many similarities to Boolean error-correcting codes [CE03]. Initial calculations suggest that there may be an analogous linear program hierarchy that upper bounds the density of *lattice packings* in $\mathbb{R}^n$ (the analog of linear codes from the Boolean setting). A distant goal is to prove new upper bounds on the density of lattice packings for large n. A possibly more tractable goal is to upper bound the density of lattice packings in a fixed dimension via computer calculation. For $n = 10$, the densest packing is conjectured is to be the Best packing, which is not a lattice packing [CS95]. Running the LP hierarchy to upper bound the density of lattice packings in $\mathbb{R}^{10}$ may be able to prove the following conjecture.

Conjecture 7.45. *The Best packing is denser than any lattice packing in* $\mathbb{R}^{10}$.

REFERENCES

[ABE$^+$05] Sanjeev Arora, Eli Berger, Hazan Elad, Guy Kindler, and Muli Safra. On non-approximability for quadratic programs. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 206–215. IEEE, 2005. 46, 86

[AH18] Albert Atserias and Tuomas Hakoniemi. Size-degree trade-offs for sums-of-squares and positivstellensatz proofs. *arXiv preprint arXiv:1811.01351*, 2018. 34

[AJT19] Vedat Levi Alev, Fernando Granha Jeronimo, and Madhur Tulsiani. Approximating constraint satisfaction problems on high-dimensional expanders. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 180–201. IEEE, 2019. 29

[AMP20] Kwangjun Ahn, Dhruv Medarametla, and Aaron Potechin. Graph matrices: Norm bounds and applications. *arXiv preprint arXiv:1604.03423*, 2020. 8, 14, 27, 67, 129

[BABB21] Enric Boix-Adserà, Matthew Brennan, and Guy Bresler. The average-case complexity of counting cliques in Erdős–Rényi hypergraphs. *SIAM Journal on Computing*, (0):FOCS19–39, 2021. 29

[BBK$^+$21a] Mitali Bafna, Boaz Barak, Pravesh K Kothari, Tselil Schramm, and David Steurer. Playing unique games on certified small-set expanders. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1629–1642, 2021. 29

[BBK$^+$21b] Afonso S Bandeira, Jess Banks, Dmitriy Kunisky, Christopher Moore, and Alex Wein. Spectral planting and the hardness of refuting cuts, colorability, and communities in random graphs. In *Conference on Learning Theory*, pages 410–473. PMLR, 2021. 126

[BHK$^+$16] Boaz Barak, Samuel B Hopkins, Jonathan Kelner, Pravesh Kothari, Ankur Moitra, and Aaron Potechin. A nearly tight sum-of-squares lower bound for the planted clique problem. In *Proceedings of the 57th IEEE Symposium on Foundations of Computer Science*, pages 428–437, 2016. xi, 3, 8, 29, 123, 125, 126, 129, 134, 136, 137, 150

[BHK21] Andreas Björklund, Thore Husfeldt, and Petteri Kaski. The shortest even cycle problem is tractable. *arXiv preprint arXiv:2111.02992*, 2021. 122

[BHKL22] Mitali Bafna, Max Hopkins, Tali Kaufman, and Shachar Lovett. High dimensional expanders: Eigenstripping, pseudorandomness, and unique games. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1069–1128. SIAM, 2022. 29

[BLM15] Charles Bordenave, Marc Lelarge, and Laurent Massoulié. Non-backtracking spectrum of random graphs: community detection and non-regular ramanujan graphs. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1347–1357. IEEE, 2015. 130

[BMR21] Jess Banks, Sidhanth Mohanty, and Prasad Raghavendra. Local statistics, semidefinite programming, and community detection. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1298–1316. SIAM, 2021. 37

[Bol02] Béla Bollobás. Evaluations of the circuit partition polynomial. *J. Combin. Theory Ser. B*, 85(2):261–268, 2002. `doi:10.1006/jctb.2001.2102`. 91

[Bor15] Charles Bordenave. A new proof of friedman's second eigenvalue theorem and its extension to random lifts. *arXiv preprint arXiv:1502.04482*, 2015. 12, 130

[BP21] Ainesh Bakshi and Adarsh Prasad. Robust linear regression: Optimal rates in polynomial time. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 102–115, 2021. 29

[BRS11] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *2011 ieee 52nd annual symposium on foundations of computer science*, pages 472–481. IEEE, 2011. 29

[BS16] Boaz Barak and David Steurer. Proofs, beliefs, and algorithms through the lens of sum-of-squares. *http://www.sumofsquares.org/public/index.html*, 2016. 29

[CE03] Henry Cohn and Noam Elkies. New upper bounds on sphere packings i. *Annals of Mathematics*, pages 689–714, 2003. 200, 206

[CG02] Fan Chung and Ronald Graham. Sparse quasi-random graphs. *Combinatorica*, 22(2):217–244, 2002. 138

[CJJ22] Leonardo Nagami Coregliano, Fernando Granha Jeronimo, and Chris Jones. A complete linear programming hierarchy for linear codes. In *Proceedings of the 13th Conference on Innovations in Theoretical Computer Science*, 2022. 4, 177, 183, 186, 187, 188, 206

[CM18] Eden Chlamtáč and Pasin Manurangsi. Sherali-adams integrality gaps matching the log-density threshold. *arXiv preprint arXiv:1804.07842*, 2018. 128

[CO05] Amin Coja-Oghlan. The lovász number of random graphs. *Combinatorics, Probability and Computing*, 14(4):439–465, 2005. 124

[COE15] Amin Coja-Oghlan and Charilaos Efthymiou. On independent sets in random graphs. *Random Structures & Algorithms*, 47(3):436–486, 2015. 124, 138

[CP20] Wenjun Cai and Aaron Potechin. The spectrum of the singular values of z-shaped graph matrices. *arXiv preprint arXiv:2006.14144*, 2020. 14, 27

[CP22] Wenjun Cai and Aaron Potechin. On mixing distributions via random orthogonal matrices and the spectrum of the singular values of multi-z shaped graph matrices. *arXiv preprint arXiv:2206.02224*, 2022. 14, 27

[CS95] John H Conway and Neil JA Sloane. What are all the best sphere packings in low dimensions? *Discrete & Computational Geometry*, 13(3):383–403, 1995. 206

[CS18] Peter J. Cameron and Jason Semeraro. The cycle polynomial of a permutation group. *Electron. J. Combin.*, 25(1):Paper No. 1.14, 13, 2018. 92

[CW04] Moses Charikar and Anthony Wirth. Maximizing quadratic programs: Extending grothendieck's inequality. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 54–60. IEEE, 2004. 46

[Del73] P. Delsarte. *An Algebraic Approach to the Association Schemes of Coding Theory*. Philips Journal of Research / Supplement. N.V. Philips' Gloeilampenfabrieken, 1973. 179, 181, 188

[DK21] Ilias Diakonikolas and Daniel M Kane. Non-gaussian component analysis via lattice basis reduction. *arXiv preprint arXiv:2112.09104*, 2021. 48

[dKPS07] Etienne de Klerk, Dmitrii V Pasechnik, and Alexander Schrijver. Reduction of symmetric semidefinite programs using the regular $*$-representation. *Mathematical programming*, 109(2):613–624, 2007. 181

[DM11] Varsha Dani and Cristopher Moore. Independent sets in random graphs from the weighted second moment method. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 472–482. Springer, 2011. 124

[DM15] Yash Deshpande and Andrea Montanari. Improved sum-of-squares lower bounds for hidden clique and hidden submatrix problems. In *Conference on Learning Theory*, pages 523–562. PMLR, 2015. 8

[DMO$^+$19] Yash Deshpande, Andrea Montanari, Ryan O'Donnell, Tselil Schramm, and Subhabrata Sen. The threshold for sdp-refutation of random regular nae-3sat. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2305–2321. SIAM, 2019. 130

[DMS17] Amir Dembo, Andrea Montanari, and Subhabrata Sen. Extremal cuts of sparse random graphs. *The Annals of Probability*, 45(2):1190–1217, 2017. 86

[DSS16] Jian Ding, Allan Sly, and Nike Sun. Maximum independent sets on random regular graphs. *Acta Mathematica*, 217(2):263–340, 2016. 124

[DX13] Feng Dai and Yuan Xu. *Approximation theory and harmonic analysis on spheres and balls*. Springer Monographs in Mathematics. Springer, New York, 2013. `doi:10.1007/978-1-4614-6660-4`. 91, 100

[DX14] Charles F. Dunkl and Yuan Xu. *Orthogonal polynomials of several variables*, volume 155 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, second edition, 2014. `doi:10.1017/CBO9781107786134`. 91

[EM98] Joanna A. Ellis-Monaghan. New results for the Martin polynomial. *J. Combin. Theory Ser. B*, 74(2):326–352, 1998. `doi:10.1006/jctb.1998.1853`. 91

[Fil16] Yuval Filmus. An orthogonal basis for functions over a slice of the Boolean hypercube. *Electron. J. Combin.*, 23(1):Paper 1.23, 27, 2016. 91

[FKP19] Noah Fleming, Pravesh Kothari, and Toniann Pitassi. Semialgebraic proofs and efficient algorithm design. *Foundations and Trends in Theoretical Computer Science*, 14(1-2):1–221, 2019. URL: `http://dx.doi.org/10.1561/0400000086`, `doi:10.1561/0400000086`. 29, 32, 34, 35

[FM17] Zhou Fan and Andrea Montanari. How well do local algorithms solve semidefinite programs? In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 604–614. ACM, 2017. 130

[FS02] Uriel Feige and Gideon Schechtman. On the optimality of the random hyperplane rounding technique for max cut. *Random Structures & Algorithms*, 20(3):403–440, 2002. 36, 85

[FT05] Joel Friedman and Jean-Pierre Tillich. Generalized Alon–Boppana theorems and error-correcting codes. *SIAM Journal on Discrete Mathematics*, 19(3):700–718, 2005. 180

[FV18] S. Friedli and Y. Velenik. *Statistical mechanics of lattice systems*. Cambridge University Press, Cambridge, 2018. 140

[Gij09] Dion Gijswijt. Block diagonalization for algebra's associated with block codes. *arXiv preprint arXiv:0910.4515*, 2009. 181, 192

[Gil52] Edgar N Gilbert. A comparison of signalling alphabets. *The Bell system technical journal*, 31(3):504–522, 1952. 179

[GJJ$^+$20] Mrinalkanti Ghosh, Fernando Granha Jeronimo, Chris Jones, Aaron Potechin, and Goutham Rajendran. Sum-of-squares lower bounds for Sherrington-Kirkpatrick via planted affine planes. In *Proceedings of the 61st IEEE Symposium on Foundations of Computer Science*. 2020. 4, 5, 8, 15, 29, 39, 41, 46, 49, 61, 62, 68, 69, 83, 87

[GM75] Geoffrey R Grimmett and Colin JH McDiarmid. On colouring random graphs. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 77, pages 313–324. Cambridge University Press, 1975. 125

[GMS12] Dion C Gijswijt, Hans D Mittelmann, and Alexander Schrijver. Semidefinite code bounds based on quadruple distances. *IEEE Transactions on Information Theory*, 58(5):2697–2705, 2012. 181, 192

[GS11] Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with psd objectives. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 482–491. IEEE, 2011. 29

[GS14] David Gamarnik and Madhu Sudan. Limits of local algorithms over sparse random graphs. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 369–376, 2014. 176

[HKP$^+$17] Samuel B Hopkins, Pravesh K Kothari, Aaron Potechin, Prasad Raghavendra, Tselil Schramm, and David Steurer. The power of sum-of-squares for detecting hidden structures. In *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science*, pages 720–731. IEEE, 2017. 44, 86

[HL18] Samuel B Hopkins and Jerry Li. Mixture models, robustness, and sum of squares proofs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1021–1034, 2018. 29

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006. 46

[Hol81] Ian Holyer. The NP-completeness of some edge-partition problems. *SIAM J. Comput.*, 10(4):713–717, 1981. doi:10.1137/0210054. 98

[Hop18] Samuel Brink Klevit Hopkins. Statistical inference and the sum of squares method. 2018. 37, 44, 86, 127

[HSS15] Samuel B Hopkins, Jonathan Shi, and David Steurer. Tensor principal component analysis via sum-of-squares proofs. In *Conference on Learning Theory*, pages 956–1006, 2015. 29

[HW20] Justin Holmgren and Alexander S Wein. Counterexamples to the low-degree conjecture. *arXiv preprint arXiv:2004.08454*, 2020. 44

[Jan97] Svante Janson. *Gaussian Hilbert spaces*, volume 129 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 1997. doi:10.1017/CBO9780511526169. 90, 91

[JH16] Cédric Josz and Didier Henrion. Strong duality in lasserre's hierarchy for polynomial optimization. *Optimization Letters*, 10(1):3–10, 2016. 34

[JP22] Chris Jones and Aaron Potechin. Almost-orthogonal bases for inner product polynomials. In *Proceedings of the 13th Conference on Innovations in Theoretical Computer Science*, 2022. 4, 87, 104

[JPR$^+$21] Chris Jones, Aaron Potechin, Goutham Rajendran, Madhur Tulsiani, and Jeff Xu. Sum-of-squares lower bounds for sparse independent set. In *Proceedings of the 62nd IEEE Symposium on Foundations of Computer Science*. 2021. 4, 8, 16, 29, 123, 135, 161, 164, 165, 171, 175

[Kar72] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972. 46

[KB19] Dmitriy Kunisky and Afonso S Bandeira. A tight degree 4 sum-of-squares lower bound for the Sherrington–Kirkpatrick hamiltonian. *arXiv preprint arXiv:1907.11686*, 2019. 48

[KKM18] Adam Klivans, Pravesh K Kothari, and Raghu Meka. Efficient algorithms for outlier-robust regression. In *Conference On Learning Theory*, pages 1420–1430. PMLR, 2018. 29

[KOS19] Pravesh Kothari, Ryan O'Donnell, and Tselil Schramm. Sos lower bounds with hard constraints: think global, act local. In *Proceedings of the 10th conference on Innovations in theoretical computer science*, 2019. 44

[KOTZ14] Manuel Kauers, Ryan O'Donnell, Li-Yang Tan, and Yuan Zhou. Hypercontractive inequalities via sos, and the frankl–rödl graph. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1644–1658. SIAM, 2014. 34

[Kun20] Dmitriy Kunisky. Positivity-preserving extensions of sum-of-squares pseudomoments over the hypercube. *arXiv preprint arXiv:2009.07269*, 2020. 29

[Lau07] Monique Laurent. Strengthened semidefinite programming bounds for codes. *Mathematical Programming*, 109:1436–4646, 2007. 181

[LRS15] James R Lee, Prasad Raghavendra, and David Steurer. Lower bounds on the size of semidefinite programming relaxations. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 567–576, 2015. 2, 44

[Mac63] Jessie MacWilliams. A theorem on the distribution of weights in a systematic code. *Bell System Technical Journal*, 42(1):79–94, 1963. 188

[Mar77] Pierre Martin. *Enumérations Eulériennes dans les multigraphes et invariants de Tutte-Grothendieck*. 1977. Ph. D. Thesis. 91

[MM09] Marc Mezard and Andrea Montanari. *Information, physics, and computation*. Oxford University Press, 2009. 204

[Mon21] Andrea Montanari. Optimization of the Sherrington–Kirkpatrick hamiltonian. *SIAM Journal on Computing*, (0):FOCS19–1, 2021. 48, 85

[MPW15] Raghu Meka, Aaron Potechin, and Avi Wigderson. Sum-of-squares lower bounds for planted clique. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 87–96, 2015. 8

[MR10] Cristopher Moore and Alexander Russell. Circuit partitions and #p-complete products of inner products. *arXiv preprint arXiv:1001.2314*, 2010. 92

[MRRW77] Robert McEliece, Eugene Rodemich, Howard Rumsey, and Lloyd Welch. New upper bounds on the rate of a code via the delsarte-macwilliams inequalities. *IEEE transactions on Information Theory*, 23(2):157–166, 1977. 177, 180, 200, 205, 206

[MRX20] Sidhanth Mohanty, Prasad Raghavendra, and Jeff Xu. Lifting sum-of-squares lower bounds: degree-2 to degree-4. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 840–853, 2020. 29, 47, 48, 49, 84

[MS16] Andrea Montanari and Subhabrata Sen. Semidefinite programs on sparse random graphs and their application to community detection. In *Proceedings of the 48th ACM Symposium on Theory of Computing*, pages 814–827, 2016. 48

[MSG72] FJ MacWilliams, NJA Sloane, and J-M Goethals. The macwilliams identities for nonlinear codes. *Bell System Technical Journal*, 51(4):803–819, 1972. 182

[MSS16] Tengyu Ma, Jonathan Shi, and David Steurer. Polynomial-time tensor decompositions with sum-of-squares. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 438–446. IEEE, 2016. 29

[NS05] Michael Navon and Alex Samorodnitsky. On Delsarte's linear programming bounds for binary codes. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 327–336. IEEE, 2005. 180

[NS09] Michael Navon and Alex Samorodnitsky. Linear programming bounds for codes via a covering argument. *Discrete & Computational Geometry*, 41(2):199–207, 2009. 180

[O'D14] Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014. 16, 126

[OVW16] Sean O'Rourke, Van Vu, and Ke Wang. Eigenvectors of random matrices: a survey. *Journal of Combinatorial Theory, Series A*, 144:361–442, 2016. 84

[OW08] Ryan O'Donnell and Yi Wu. An optimal sdp algorithm for max-cut, and equally optimal long code tests. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 335–344, 2008. 36

[Pan21] Shuo Pang. Sos lower bound for exact planted clique. In *36th Computational Complexity Conference (CCC 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021. 126, 176

[Par79] Giorgio Parisi. Infinite number of order parameters for spin-glasses. *Physical Review Letters*, 43(23):1754, 1979. 45, 48

[PR20] Aaron Potechin and Goutham Rajendran. Machinery for proving sum-of-squares lower bounds on certification problems. *arXiv preprint arXiv:2011.04253*, 2020. 8, 29, 41, 42, 134

[Rag08] Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 245–254, 2008. 2

[Raj22] Goutham Rajendran. *Nonlinear Random Matrices and Applications to the Sum of Squares Hierarchy*. 2022. Ph. D. Thesis. 5

[Rom05] Steven Roman. *The umbral calculus*. Springer, 2005. 106

[RSS18] Prasad Raghavendra, Tselil Schramm, and David Steurer. High dimensional estimation via sum-of-squares proofs. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages 3389–3423. World Scientific, 2018. 29

[RT20] Goutham Rajendran and Madhur Tulsiani. Concentration of polynomial random matrices via efron-stein inequalities. Manuscript, 2020. 15, 126

[RV17a] Mustazee Rahman and Balint Virag. Local algorithms for independent sets are half-optimal. *The Annals of Probability*, 45(3):1543–1577, 2017. 125

[RV17b] Mustazee Rahman and Balint Virag. Local algorithms for independent sets are half-optimal. *The Annals of Probability*, 45(3):1543–1577, 2017. 176

[RW97] Gian-Carlo Rota and Timothy C Wallstrom. Stochastic integrals: a combinatorial approach. *The Annals of Probability*, 25(3):1257–1283, 1997. 76

[RW17] Prasad Raghavendra and Benjamin Weitz. On the bit complexity of sum-of-squares proofs. *arXiv preprint arXiv:1702.05139*, 2017. 30

[Sam21] Alex Samorodnitsky. One more proof of the first linear programming bound for binary codes and two conjectures. *arXiv preprint arXiv:2104.14587*, 2021. 180, 193

[Sch79] Alexander Schrijver. A comparison of the Delsarte and Lovász bounds. *IEEE Transactions on Information Theory*, 25(4):425–429, 1979. 180

[Sch05] Alexander Schrijver. New code upper bounds from the Terwilliger algebra and semidefinite programming. *IEEE Transactions on Information Theory*, 51(8):2859–2866, 2005. 181, 192

[Sta12] Richard P. Stanley. *Enumerative combinatorics. Volume 1*, volume 49 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, second edition, 2012. 117, 118, 119

[Tal06] Michel Talagrand. The Parisi formula. *Annals of mathematics*, pages 221–263, 2006. 45, 48

[Tao12] Terence Tao. *Topics in random matrix theory*, volume 132. American Mathematical Soc., 2012. 12

[Val19] Frank Vallentin. Semidefinite programming bounds for error-correcting codes. *arXiv preprint arXiv:1902.01253*, 2019. 184

[Var57] Rom Rubenovich Varshamov. Estimate of the number of signals in error correcting codes. *Docklady Akademii Nauk, SSSR*, 117:739–741, 1957. 179

[Via17] Maryna S Viazovska. The sphere packing problem in dimension 8. *Annals of Mathematics*, pages 991–1015, 2017. 200

[Wei20] Alexander S Wein. Optimal low-degree hardness of maximum independent set. *arXiv preprint arXiv:2010.06563*, 2020. 125

[Wig93] Eugene P Wigner. Characteristic vectors of bordered matrices with infinite dimensions I. In *The Collected Works of Eugene Paul Wigner*, pages 524–540. Springer, 1993. 84

[Wor95] Nicholas C Wormald. Differential equations for random processes and random graphs. *The annals of applied probability*, pages 1217–1235, 1995. 125

[ZSWB21] Ilias Zadik, Min Jae Song, Alexander S Wein, and Joan Bruna. Lattice-based methods surpass sum-of-squares in clustering. *arXiv preprint arXiv:2112.03898*, 2021. 44, 48