On the Expansion of Graphs

Pedro Paredes

CMU-CS-22-136

August 2022

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:

Ryan O'Donnell, Chair
Anupam Gupta
Pravesh Kothari
Luca Trevisan, Bocconi University
Nikhil Srivastava, UC Berkeley

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

This research was sponsored by the National Science Foundation under award numbers CCF-171606 and CCF-1909310, and the U.S. Army Research Office under award number W911NF2110001.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: Spectral Graph Theory, Expander graphs, Pseudorandomness

To my mother (Emília) and my little brother (Jibu)

Abstract

A popular way of analyzing a graph is through spectral properties of its associated matrices, such as the adjacency matrix or the Laplacian matrix. This type of analysis has produced several insights with practical applications in diverse areas, including internet search, clustering and segmentation of data and many more. From a theoretical perspective, spectral graph theory is such a fundamental tool that its applications span virtually the whole field of theoretical computer science.

One of the many successes of this area is the notion of *graph expansion*. A graph is expanding if it is simultaneously sparse and highly connected (meaning that we need to remove a lot of edges to disconnect a large part of the graph.) Since being defined in the '70s, *expander graphs* have spawned a lot of research with many applications in mathematics, computer science and even physics.

This thesis attempts to further the study of these objects by focusing on three fundamental questions:

- **How can we construct explicit (i.e. deterministically and efficiently) expanding graphs?** We devised an explicit construction of nearly optimal expanding regular graphs of all degrees. We also showed how to use this result to obtain nearly optimal expanding graphs with high girth (i.e. that do not contain small cycles).
- **What is the expansion of random graphs drawn from different distributions?** We analyzed the expansion of several types of different random graph distributions based on graphs products, like random additive lifts and random abelian lifts.
- **How can we leverage expansion in other domains?** We showed how to analyze the SDP value of a family of random constraint satisfaction problems (CSPs) and also how to construct explicit nearly linear distance quantum low density parity check (LDPC) and quasi-cyclic LDPC error correcting codes in polynomial time.

Acknowledgments

If you are reading this then you either expect to see your name on this list or you are someone I meet in the future that is curious to see what I wrote. For the former, I really want to thank you for everything and I apologize if you don't find your name on the following list. For the latter, hello! I can't wait to meet you in the future.

When I joined CMU in 2017 to start my PhD I knew I wanted to do something theoretical and "mathy", but I was completely unprepared to do so. I lacked the knowledge of most basic concepts in theoretical computer science that most of the CMU undergrads interested in the area knew well (as I painfully found out when I took my first class). As I was learning what the professors here were working on I got really interested in the work that my now advisor Ryan O'Donnell was doing. To this day I still wonder why he decided to take me, a clueless first year, as his student, but I am deeply grateful that he did so. Ryan tolerated my ineptitude and guided me to finding an area of research that I really enjoy and am able to do work on. I learned so much more than I thought possible thanks to Ryan. I am also so thankful for all the support and motivation, I am proud to call Ryan my friend and I hope we go back to Heinz Field (my favorite place in Pittsburgh) to watch the Steelers.

Like any good PhD student, I thought about quitting multiple times. My work was never going well and I didn't think I was good enough to be a researcher. I was lucky to have many collaborators that made the process of failing tolerable and the successes fun. I have to especially thank Sidhanth Mohanty, without whom I don't think I would have been able to publish my first paper, which made me believe I could do it again. I am also really grateful to Ainesh Bakshi, who gave me so much advice, heard me whine about grad school multiple times and so much more. He was probably the most important person not named Ryan to my academic career. I also want to thank all of the amazing people that were my collaborators during my PhD: Ainesh Bakshi, Timothy Chu, Fernando Granha Jeronimo, Theo McKenzie, Tushant Mittal, Sidhanth Mohanty, Ryan O'Donnell, Kevin Pratt, Rocco Servedio, Li-Yang Tan, Luca Trevisan, Madhur Tulsiani and Xinyu Wu. Additionally, I am thankful to all the other people that gave me academic advice and supported me: Anupam Gupta, Bernhard Haeupler, Pravesh Kothari, Danny Sleator, Nikhil Srivastava and Goran Žužić. I also owe a lot to the CS Department at CMU — faculty, fellow students and administration (especially Deb and Catherine!).

Before I started my PhD I had a very different life back in my home country of Portugal. I wouldn't be here if it weren't for the people I met there. I want to especially thank Pedro Ribeiro, my advisor throughout high school and my undergraduate, who introduced me to computer science and the world of research. I am also grateful to all the professors and teachers that taught me so much even though I was an arrogant student most of the time. And a big thank you to all my friends, I can't name you all but I want to name a few of you: Laura, Pedro Teles, Michel, David, Francisco, Rodrigo, Ramos, Pires, Filipe, Duarte, Patrick, Alberto, Claúdia, Ricardo, Castanheira, Miguel and Kevin.

I was so so lucky to have had such amazing friends during my time in Pittsburgh. Once again I can't name you all but I'll try to name a few: Ainesh Bakshi, Michaela Barry, David Bernal, Rodrigo Bernardo, Vijay Bhattiprolu, Emily Black, Calyl, Tim Chu, Marina DiMarco, Magdalen Dobson, Aymeric Fromherz, Sydney Gibson, Ellis Hershkowitz, Raj Jayaram ♡, Kai Jayaram, Ryan Kavanagh, Greg Kehne, Klas Leino, Roie Levin, Jason Li, Peter Manohar, Sidhanth Mohanty, Luís Oliveira, Filipe Peres, Kevin Pratt, Nic Resch, João Ribeiro, Elena Salas, David Wajc, Maya Shen, Rui Silva, Alex Wang, Sam Westrick, Xinyu Wu, Sofia Gómez, Jeff Xu, Goran Žužić. You all made me enjoy life and grow immensely as person. I am who I am because of you. I also want to especially thank Emily Black and Laura Sobral, who helped me survive the toughest days of the COVID-19 pandemic and have been my support network for a long time.

I could never forget to thank my family, who I had to abandon to start a new life in a new continent. My mother Emília and my brother Gonçalo (to whom I dedicate this thesis), my aunt Bina, my grandma Celina, my uncle Paulo and my aunt Nani, my cousins Paula, Zé e Inês, my aunt Nelita and uncle António, my cousins Inês and Rodrigo and all the kids, my "second mom" Cristina and all the Simões, my brother's partner Viktoriya. I also want to mention my grandpa Paredes, who passed away before I started college, and my grandma Mia, who passed away while I was doing my PhD, I'll never forget you. I further have to thank my "American family", the Wellners: Linda, Pierre, Jules, Tristan, Emily and Lucie, Magdeleine and Marcel.

Finally, I thank the two most important people in my life: Horace, the best kitty in the world, who laid next to me while I wrote most of this thesis and helped me proofread it; and my partner Zoe Wellner, who showed me happiness exists and supported me immensely through so much.

Thank you all.

Contents

Chapter 1

Introduction

The subject of this thesis is the *expansion* of graphs. To motivate this concept, image that some company has a big data center with n servers that need to compute something and transmit data between themselves. To do so, the company wants to connect the different servers (e.g. with physical wires) so that every pair of servers can communicate, potentially by transmitting information through other servers. At the same time, it is important that the network of servers is *robust*, which for our purposes means that if any small percentage of the wires are disconnected, most of the servers will still be able to communicate.

An immediate answer to this problem is to just connect every single pair of servers, it is impossible to do better since these are all the possible connections we can place. However, each wire we place is expensive, so what if we want to minimize the number of connections in this network of servers? To summarize, we want a network that is simultaneously sparse and highly connected. This is exactly what an *expander graph* is, i.e. a graph with good expansion

Let us now consider a different more abstract problem. Suppose we want to build an undirected graph on n vertices. After we build this graph we are going to take a random walk on it, that is, we start at an arbitrary vertex and then pick one of its neighbors uniformly at random and walk to it. We then pick another neighbor uniformly at random and we repeat this procedure until we have visited the whole graph. Our goal is to pick a graph that minimizes the expected number of steps we have to take before we visit all of the vertices of the graph.

It is not as immediate as before, but we can show that again the complete graph is the best graph. But, again, what if we want a small number of edges (i.e. a sparse graph)? Intuitively we want a sparse graph where we never get "stuck", meaning that the probability of revisiting a vertex is as small as possible as we keep walking. One can show that this is related to being highly connected and so (perhaps surprisingly) the solution to this second problem is again an expander graph.

These two examples show two applications of expander graphs, but they also illustrate how diverse they can be. Our first example is a purely combinatorial problem that concerns connectedness in graphs. The second example is a probabilities problem on the expectation of a random variable. This versatility of expanders explains why they are so ubiquitous, since they can find applications in so many domains. Furthermore, it also let us approach problems in expanders from different perspectives and we are free to pick whichever one makes the problem easier to tackle.

The history of the study of expansion started in the '70s, when they were first defined and their existence was showed. Since the, there has been an extraordinary amount of research that continues to this day and new applications continue to be found in not only computer science, but mathematics and physics.

In this thesis we will further the study of expander graphs by proposing new constructions of expander graphs, by analyzing the expansion of random graphs and by giving applications of these to theoretical computer science.

1.1 A (formal) journey through the world of expanders

Consider an undirected n-vertex multigraph (self loops and multiple edges are allowed) $G = (V(G), E(G))$, $|V(G)| = n$ and for $v \in V(G)$ let's denote by $N(v)$ the set of neighbors of v. We say that G is a *d-regular graph* if all vertices have degree exactly d, meaning that for all $v \in V(G)$ we have $|N(v)| = d$. For sets of vertices $S, T \subseteq V$ we denote by $E(S,T)$ the set of edges that have one endpoint in S and one endpoint in T, formally $E(S,T) = \{(u,v) | u \in S, v \in T, (u,v) \in E(G)\}$. We also denote by $\overline{S} = V(G) \setminus S$ the complement of S. The *edge boundary* of a set S, denoted ∂S, is defined as the set of edges with one endpoint in S and one endpoint outside of S, or formally $\partial S = E(S, \overline{S})$. Here is one formal way to define expansion:

Definition 1.1.1 (Edge Expansion Ratio)**.** The *edge expansion ratio* of G, denoted $h(G)$, is defined as:

$$h(G) = \min_{\substack{S \in V(G) \\ |S| \leqslant \frac{n}{2}}} \frac{|\partial S|}{|S|}.$$

Note that a disconnected graph has edge expansion ratio of 0, since if we take S to be the smallest connected component of the graph (which must contain at most half the vertices) the boundary of such set is empty. It is also easy to see that if a graph has maximum degree Δ then the edge expansion ratio is at most Δ, since for a set of vertices S the maximum number of edges with one endpoint in S is at most $\Delta \cdot |S|$.

We can study the edge expansion ratio of any graph, but as we will see later, it is especially interesting to look at the case of sparse graphs. To do so we will consider d-regular graphs, since if we think of d as a constant but n going to infinity, then the resulting family of graphs is sparse. We choose to fix the degree of all vertices because on one hand this will introduce extra constraints to the problems we study (and thus, it will only make them harder to solve), but on the other hand it is nicer to analyze the regular case. Additionally, there are many theoretical applications where fixing the degree is actually important. So this motivates the following definition:

Definition 1.1.2 (Family of Edge Expander Graphs)**.** An infinite sequence of d-regular graphs $\{G_i\}_{i \in \mathbb{N}}$ of vertex set size increasing with i is said to be a *family of edge expander graphs* if there is a constant $\varepsilon > 0$ such that $h(G_i) \geqslant \varepsilon$ for all i.

This definition is motivated by our initial informal description of expanders as "sparse and highly connected". The sparseness comes from our use of d-regular graphs. As for being highly

connected, notice that if we want to disconnect a set S of vertices from the rest of the graph we need to remove at least $\varepsilon|S|$ edges.

Let's now consider a different way of characterizing expansion, through spectral properties of graphs. The adjacency matrix A_G of G is the matrix with rows and columns indexed by the vertices of G, where for $u, v \in V(G)$, $(A_G)_{uv}$ is equal to the number of occurrences of the edge $\{u, v\}$ in $E(G)$. Since we are assuming that G is undirected, A_G is a real symmetric matrix and thus the spectral theorem tells us that there are n real *eigenvalues* with corresponding orthogonal *eigenvectors*. We order the eigenvalues and denote them by $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_n$. The largest eigenvalue λ_1 is known as the *trivial eigenvalue*, which is always equal do d for regular graphs.

Definition 1.1.3 (Spectral Expansion)**.** The *spectral expansion* of G, denoted $\lambda(G)$, is defined as:

$$\lambda(G) = \max\{\lambda_2, |\lambda_n|\}.$$

And, as we did before, we can define a family of spectral expander graphs.

Definition 1.1.4 (Family of Spectral Expander Graphs)**.** An infinite sequence of d-regular graphs $\{G_i\}_{i\in\mathbb{N}}$ of vertex set size increasing with i is said to be a *family of spectral expander graphs* if there is a constant $\delta > 0$ such that $h(G_i) \leqslant (1-\delta)d$ for all i.

The aforementioned properties of A_G imply that $\lambda(G)$ is a real number between 0 and d. If the graph G is disconnected then it is known that $\lambda_2 = d$, which means $\lambda(G) = d$. Conversely, if G is a complete graph then all eigenvalues except λ_1 are -1, so $\lambda(G) = 1$. This suggests a similar behavior to the edge expansion ratio, where if a graph is disconnected then $\lambda(G)$ is large (and $h(G)$ is small) and if a graph is highly connected then $\lambda(G)$ is small (and $h(G)$ is large), hence why $\lambda(G)$ is known as spectral expansion. This connection can be made quantitative through what is known as *Cheeger's Inequality*, which was originally proved by Dodziuk [Dod84] and independently by Alon and Milman [AM85].

Theorem 1.1.5 (Cheeger's Inequality)**.**

$$\frac{d-\lambda_2}{2} \leqslant h(G) \leqslant \sqrt{2d(d-\lambda_2)}.$$

Another way to interpret how $\lambda(G)$ governs the expansion of a graph is through the *Expander Mixing Lemma*, which was first proved by Alon and Chung [AC88].

Lemma 1.1.6 (Expander Mixing Lemma)**.** *For all $S, T \subseteq V$':*

$$\left||E(S,T)| - \frac{d|S||T|}{n}\right| \leqslant \lambda(G)\sqrt{|S||T|}.$$

We can interpret this lemma in two ways. First, by considering $T = \overline{S}$ we recover another connection between $\lambda(G)$ and $h(G)$. Second, the quantity $d|S||T|/n$ is the expected number of edges between S and T in a random graph of density d/n. So this lemma tells us that in a spectral expanding graph all pairs of sets of vertices have a number of crossing edges close to what is expected in a random graph. This property motivates calling expander graphs *pseudorandom* and indeed the Expander Mixing Lemma finds many applications in the theory of pseudorandomness.

A natural question that follows from the above is: how small can $\lambda(G)$ be? This fundamentally asks what is the optimal spectral expander. The answer is given by the well-known *Alon-Boppana bound*, which was shown in a series of works [Alo86, Nil91, Fri93], that says:

Theorem 1.1.7 (Alon-Boppana Bound)**.**

$$\lambda_2 \geqslant 2\sqrt{d-1} - O\left(\frac{1}{\log^2 n}\right).$$

This shows that $2\sqrt{d-1}$ is essentially a lower bound to the spectral expansion and hence it leads to the following definition of optimal spectral expanders:

Definition 1.1.8 (Ramanujan Graphs)**.** A d-regular (multi)graph G is called *(two-sided) Ramanujan* whenever $\lambda(G) \leqslant 2\sqrt{d-1}$. When we merely have $\lambda_2 \leqslant 2\sqrt{d-1}$, we call G *one-sided Ramanujan*.

Now the question of the existence of *explicit constructions* of such graphs arises. By explicit we mean that there is a deterministic and efficient algorithm to generate such a graph. However, we can define efficiency in different ways.

Definition 1.1.9 (Explicit Constructions)**.** We define the following three notions of explicitness:

- An algorithm is *weakly explicit* if given integers n and d, it generates an n-vertex d-regular Ramanujan graph in time polynomial in n.
- An algorithm is *strongly explicit* if given integers n and d, it generates a representation A of an n-vertex d-regular Ramanujan graph such that: when given as input a vertex $v \in \{1, \ldots, n\}$ and a neighbor $i \in \{1, \ldots, d\}$, there is an algorithm that computes the ith neighbor of v in $\mathrm{polylog}(n)$ time.
- An algorithm is *probabilistically strongly explicit* if given integers n and d and a seed $s \in \{0,1\}^{O(\log n)}$, it generates a representation A of an n-vertex d-regular graph that is Ramanujan with high probability over the choice of seed s and such that: when given as input a vertex $v \in \{1, \ldots, n\}$ and a neighbor $i \in \{1, \ldots, d\}$, there is an algorithm that computes the ith neighbor of v in $\mathrm{polylog}(n)$ time.

Our discussion so far justifies the first fundamental question that this thesis is centered around:

Question 1. *How can we construct explicit expanding graphs?*

1.1.1 Constructing expanding graphs

Margulis [Mar73] was the first to provide an explicit expander family; a slight variant of it, which is 8-regular, was shown [GG81] to have $\lambda \leqslant 5\sqrt{2} \approx 7.1$ (see [HLW06]).

When $d-1$ is an odd prime, Ihara [Iha66] (implicitly, cf. [Cla06]) and Lubotzky–Phillips–Sarnak [LPS88] and Margulis [Mar88] (independently) showed how to explcititly construct Ramanujan graphs. The $d-1=2$ case was construced by Chiu [Chi92]. The general prime power case mentioned below is due to Morgenstern [Mor94]. For extensions to general d where the eigenvalue bound depends on the number of distinct prime divisors of $d-1$, see [Piz90, Cla06].

Theorem 1.1.10. *([Mor94].) For any $d \geqslant 3$ with $d-1$ a prime power, there is a strongly explicit family of d-regular Ramanujan graphs.*

For all other values of d — e.g., for $d=7$ — it is unknown if infinite families of d-regular Ramanujan graphs exist (but see Theorem 1.1.15 below for the one-sided bipartite case).

A natural question then is whether, for every d, one can achieve *explicit* graph families that are "ε-near-Ramanujan", meaning families that have $\lambda(G) \leqslant 2\sqrt{d-1} + \varepsilon$. In their work introducing the *zig-zag product*, Reingold–Vadhan–Wigderson [RVW02] asked whether explicit families could at least reach a bound of $O(\sqrt{d})$; towards this, their work gave strongly explicit families with $\lambda(G) \leqslant O(d^{2/3})$. By extending their approach, Ben-Aroya and Ta-Shma reached $d^{1/2+o(1)}$:

Theorem 1.1.11. *([RVW02, BT11].) There are strongly explicit families of d-regular multigraphs G satisfying the bound $\lambda(G) \leqslant \sqrt{d} \cdot 2^{O(\sqrt{\log d})}$.*

Bilu and Linial [BL06] got even closer to $O(\sqrt{d})$, using a new approach based on random *lifts* that will prove important to us. Their graph families are not strongly explicit, although Bilu–Linial point out they are at least probabilistically strongly explicit.

Theorem 1.1.12. *([BL06].) There are explicit families of d-regular multigraphs G satisfying the bound $\lambda(G) \leqslant \sqrt{d} \cdot O(\log^{1.5} d)$.*

Due to their asymptotic-in-d nature, neither of Theorems 1.1.11 and 1.1.12 gives much help for specific small values of d not covered by Morgenstern, such as $d = 7$. In such cases, one can use a simple idea due to Cioabă and Murty [CM08] (cf. [dlHM06]): take a prime (or prime power) $q < d - 1$, form a $(q+1)$-regular Ramanujan graph, and then add in $d - q - 1$ arbitrary perfect matchings. It is shown in [CM08] that each perfect matching increases $\lambda(G)$ by at most 1. Hence:

Theorem 1.1.13. *([CM08].) For any $d \geqslant 3$, there is a strongly explicit family of d-regular multigraphs with $\lambda(G) \leqslant 2\sqrt{d-1} + \mathrm{gap}(d)$, where $\mathrm{gap}(d)$ denotes the least value g such that $d - 1 - g$ is a prime (power). One can bound $\mathrm{gap}(d)$ by $O(\log^2 d)$ under Cramér's conjecture, by $O(\sqrt{d} \log d)$ under the Riemann Hypothesis, or by $O(d^{.525})$ unconditionally.*

For example, this gives strongly explicit 7-regular multigraphs with $\lambda(G) \leqslant 2\sqrt{5} + 1 < 5.5$. For comparison, the Ramanujan bound is $2\sqrt{6} < 4.9$.

A similar simple idea was pointed out to us by Noga Alon, who first mentioned it in several lecture notes in the 90s. Given two n-vertex graphs G_1 and G_2 respectively d_1-regular and d_2-regular, which satisfy $\lambda(G_1) \leqslant \lambda_1$ and $\lambda(G_2) \leqslant \lambda_2$, the edge disjoint union of the two forms an n-vertex graph which is $d_1 + d_2$-regular and has $\lambda \leqslant \lambda_1 + \lambda_2$. With this in mind, let $d_1 = d$ and take the largest prime p_1 with $p_1 + 1 \leqslant d_1$, such that there are Ramanujan graphs with degree p_1 (as given by [LPS88]). Now, put $d_2 = d_1 - p_1 - 1$ and find the largest prime p_2 with $p_2 + 1 \leqslant d_2$, such that there are Ramanujan graphs with degree p_2. Repeat this procedure several times and take the edge-disjoint union of each produced graph. This results in a d-regular graph with $\lambda \leqslant (2 + o_d(1))\sqrt{d}$. This recently appeared in print in [Alo21]. Note that this is not strongly explicit since it requires finding large primes, Alon addressed this issue in [Alo21] to obtain the following:

Theorem 1.1.14. *([Alo21].) For any $d \geqslant 3$, there is a strongly explicit family of d-regular graphs with $\lambda(G) \leqslant (2 + o_d(1))\sqrt{d}$.*

Finally, Marcus–Spielman–Srivastava [MSS15a, MSS15b] recently introduced the *Interlacing Polynomials Method* and used it to show that *one-sided bipartite* Ramanujan graphs exist for all $d \geqslant 3$ and all even n. Their proof was merely existential, but Cohen [Coh16] was able to make it explicit (though not strongly so):

Theorem 1.1.15. *([MSS15a, MSS15b, Coh16].) For any $d \geqslant 3$, there is an explicit family of one-sided bipartite, d-regular, Ramanujan multigraphs.*

As mentioned, this theorem gives an n-vertex graph for every even n, which is slightly better than all other results mentioned in this section, which merely give graphs for a dense sequence of n's (typically, a sequence n_j with $n_{j+1} - n_j = o(n_j)$). Also, as pointed out to us by Nikhil Srivastava, pairing left and right vertices in the construction from Theorem 1.1.15 and merging them gives "twice-Ramanujan" graphs of every even degree; i.e., $2d$-regular graphs for all $d \geqslant 3$ with $\lambda(G) \leqslant 4\sqrt{d-1}$. We include a short proof here: let $\widetilde{A} = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$ be the adjacency matrix of a d-regular bipartite Ramanujan graph. Then $A + A^T$ is the adjacency matrix of the merged graph. For any x orthogonal to $\vec{1}$, $(A + A^T)x = \begin{bmatrix} \mathbb{1} & \mathbb{1} \end{bmatrix} \widetilde{A} \begin{bmatrix} x \\ x \end{bmatrix}$. Thus $\left\|(A + A^T)x\right\| \leqslant \sqrt{2} \cdot \left\| \widetilde{A} \begin{bmatrix} x \\ x \end{bmatrix} \right\|$, where $\vec{1}$ is the all-ones vector and $\mathbb{1}$ the identity matrix. Since $\begin{bmatrix} x \\ x \end{bmatrix}$ is orthogonal to both $\begin{bmatrix} \vec{1} \\ \vec{1} \end{bmatrix}$ and $\begin{bmatrix} \vec{1} \\ -\vec{1} \end{bmatrix}$, we have $\left\| \widetilde{A} \begin{bmatrix} x \\ x \end{bmatrix} \right\| \leqslant 2\sqrt{d-1}\sqrt{2}\|x\|$. One can then conclude that $\|(A + A^T)x\| \leqslant 4\sqrt{d-1}\|x\|$.

One can then obtain $(2d+1)$-regular graphs with $\lambda(G) \leqslant 4\sqrt{d-1}+1$ by adding an arbitrary perfect matching via the result of [CM08].

We summarize all these results in Section 1.1.1.

Table 1.1: Summary of expander construction prior to this thesis.

Who?	**Which *d*?**	**Eigenvalue bound**	**2-sided?**	**Strongly explicit?**	**Always simple?**	**# vertices given *n***
[Iha66, LPS88, Mar88, Chi92, Mor94]	prime power + 1	$2\sqrt{d-1}$	✓	✓	✓	$n(1+o(1))$
[Piz90, Cla06]	any d	$2^{\mathrm{om}(d-1)}\sqrt{d-1}$ *	✓	✓	✗	$n(1+o(1))$
[RVW02, BT11]	any d	$\sqrt{d} \cdot 2^{O(\sqrt{\log d})}$	✓	✓	✗	$\Theta(n)$
[dlHM06, CM08]	any d	$\begin{cases} 2\sqrt{d-1} + O(\log^2 d) & \dagger \\ \sqrt{d} \cdot O(\log d) & \ddagger \\ O(d^{.525}) \end{cases}$	✓	✓	✓§	$n(1+o(1))$
[BL06]	any d	$\sqrt{d} \cdot O(\log^{1.5} d)$	✓	✗¶	✓	$n(1+o(1))$
[Alo21]	any d	$(2+o_d(1))\sqrt{d}$	✓	✓	✓	$n(1+o(1))$
[MSS15a, MSS15b, Coh16]	any d	$2\sqrt{d-1}$	✗	✗	✗	$2\lceil n/2 \rceil$

* In the this entry we have written $\mathrm{om}(d-1)$ for the number of distinct prime divisors of $d-1$. Thus [Cla06] generalizes the preceding "prime power + 1" entry of [Mor94]. Also, $2^{\mathrm{om}(d-1)}$ is at most $2^{O(\log d/\log\log d)} = d^{o(1)}$ for all d, and is $(2+o(1))^{\ln\ln d} = O(\log^{0.7} d)$ for "most" d.

† Assuming Cramér's conjecture. ‡ Assuming the Riemann Hypothesis.

§ The construction can be made simple at the expense of making it not strongly explicit.

¶ The construction is probabilistically strongly explicit.

One of the main contributions of this thesis is a probabilistically strongly explicit construction of nearly optimal expanding regular graphs. In particular, we construct near-Ramanujan graphs (i.e. graphs that satisfy $\lambda(G) \leqslant 2\sqrt{d-1} + \varepsilon$, for positive constant ε) of arbitrary degree. To establish this result, we showed how to find expanding *two-lifts*. These are a type of graph product that doubles the number of vertices of a regular graph while preserving its regularity. Additionally, it "mildly" scrambles the edges of the original graph, so that a random application

of this graph product retains the spectral properties of the base graph with high probability. The implication of this to the construction of near-Ramanujan graphs is that we can do so by starting with a small base graph with nice spectral properties and then repeatedly lift it until we obtain a large expanding graph.

Besides constructing expanding graphs, it is also of interest to construct expanding graphs that additionally have some other property. Let us consider the *girth* of a graph, that is the length of the shortest cycle of a graph. Expanding graphs that also have high girth, meaning that do not contain any short cycles, have many applications in different areas. One example of such is in coding theory, where expander graphs with high-girth help constructing certain error corrector codes that can be efficiently decoded. Motivated by the previous, another contribution of this thesis is the construction of near-Ramanujan graphs that also have high girth.

An additional property that is interesting to combining with expansion is the existence of certain *symmetries*. Informally, we say that G has symmetries of H if $H \subseteq \mathrm{Aut}(G)$, where $\mathrm{Aut}(G)$ denotes the group of all graph isomorphisms to itself. As we will see in Chapter 7, constructing expander graphs with given symmetries has many applications. With this motivation in mind, in we study a generalization of the two-lift graph product based on groups and we describe explicit constructions of expanders obtained via *abelian lifts*.

Throughout this thesis as we prove these results we will see that we apply similar ideas to show that the graph products of interest are expanding. This "core proof template" was introduced and then refined and repeatedly applied. To ease the reading of the main technical sections, we provide in Chapter 2 a mostly self-contained proof of the simplest application of this proof template, using ideas from all the pretty much all chapters of this thesis.

A natural follow up to the question of this section is to ask how is expansion distributed over random graphs of different distributions. This is exactly what this thesis' second fundamental question asks.

Question 2. *What is the spectral expansion of random graphs drawn from different distributions?*

1.1.2 The expansion of random graphs

The most well-known result of this nature is related to the behavior of uniformly random regular graphs. Alon [Alo86] conjectured that a random n-vertex d-regular graph $\boldsymbol{G}$ has $\lambda(\boldsymbol{G}) \leqslant 2\sqrt{d-1}+o_n(1)$ with high probability, and this was proven two decades later by Friedman [Fri08] and later a simpler proof was given by Bordenave [Bor19].

Theorem 1.1.16 ([Fri08].)**.** *Fix any $d \geqslant 3$ and $\varepsilon > 0$ and let $\boldsymbol{G}$ be a uniformly random d-regular graph. Then*

$$\mathbf{Pr}\Big[\lambda(\boldsymbol{G}) \leqslant 2\sqrt{d-1}+\varepsilon\Big] \geqslant 1-o_n(1).$$

In fact [Bor19], $\boldsymbol{G}$ achieves the subconstant $\varepsilon = \widetilde{O}(1/\log^2 n)$ with probability at least $1-1/n^{.99}$.

We will see in this thesis how to achieve similar results to graphs drawn from very different looking distributions. In fact, most of the construction results mentioned in the previous subsection were obtained by showing that random graphs drawn from a certain distribution are expanding with high probability and then we show how to derandomize those results. So to find expanding two-lifts we first show that uniformly random two-lifts of a graph are expanding and

then we show how to find one by derandomizing this result. The same applies to the result on abelian lifts.

We also show a generalization of Theorem 1.1.16 to the case of *additive lifts*. These are yet another generalization of two-lifts that have applications in the theory of random constraint satisfaction problems.

Given all of the above, the final fundamental question is:

Question 3. *How can we leverage expansion in other domains?*

1.1.3 Applications of expanders

It would be impossible to summarize all of the applications of expander graphs. We recommend the surveys of Hoory-Linial-Wigderson [HLW06] and Kowalski [Kow19] for a comprehensive list of applications and connections of Ramanujan graphs and expanders to computer science and mathematics.

Our contributions focused on two areas: random constraint satisfaction problems and coding theory.

Random constraint satisfaction problems Refutation of constraint satisfaction problems (CSPs) is a fundamental problem in complexity theory. Given a CSP formula, refutation is the task of providing a proof that no assignment achieves some larger value. In the theory of algorithms and complexity, the most difficult instances of a given CSPs are arguably random (sparse) instances. Indeed, the assumed intractability of random CSPs underlies, for example, various cryptographic proposals for one-way functions [Gol00, JP00].

Given, say, a random Max-Cut instance of average degree d, its optimum value is (whp) concentrated around $\frac{1}{2} + f(d)$, where f is a certain function of d. However, the most efficient algorithms we know can only find (whp) cuts of value approximately $\frac{1}{2} + .83f(d)$ the optimal one. This suggests an "information-computation" gap. One way to study this is through the behavior of semidefinite programming (SDP) relaxations, which are the most popular and successful approaches to refuting CSPs. Given an instance of a CSP, we call the exact threshold result for when the natural SDP algorithm is able to certify unsatisfiability the *SDP value of the instance*.

In this thesis we determined the SDP value of random instances of certain kinds of constraint satisfaction problems, which are known as "two-eigenvalue 2CSPs". These are CSPs where each clause can be described by a graph where each vertex represents a variable and each edge is an XOR constraint between two variables, and such that the spectrum of the adjacency matrix of the graph only contains two distinct eigenvalues. This includes multiple famous CSPs families like the NAE-3SAT, the SORT_4 and the Forrelation_k CSPs.

Coding theory and expansion The connections between graphs and error correcting codes have a rich history. Low-density parity check (LDPC) codes were first introduced by Gallager [Gal62] in the '60s and are one of the most popular classes of classical error-correcting codes, both in theory and in practice. These are linear codes that are defined through bipartite graphs, where one set of vertices represents bits and the other set of vertices represents constraints. The

use of expansion in analyzing these codes was first described by Sipser and Spielman [SS96], who defined *expander codes*.

In the quantum world the role of error correcting codes is even more important. Quantum systems are very fragile and prone to decoherence, so it is important to be able to detect and correct errors.

In this thesis we show how to obtain a family of explicit quantum code of almost linear distance (and also in a wide range of parameters). As a corollary, we also obtain efficient constructions of a different family of classical codes. These constructions are achieved using the aforementioned generalization of lifts based on groups.

1.2 Outline of this thesis

Throughout this thesis we will attempt to answer the three fundamental questions in different ways. Here is a summary of our contributions:

- To answer Question 1, we devise an explicit construction of nearly optimal expanding regular graphs of all degrees. We also show how to use this result to obtain nearly optimal expanding graphs with high girth (i.e. that do not contain small cycles).
- To answer Question 2, we analyze the expansion of several types of different random graph distributions based on graphs products, like random additive lifts and random abelian lifts.
- To answer Question 3, we show how to analyze the SDP value of a family of random constraint satisfaction problems (CSPs) and also show how to construct explicit nearly linear distance quantum low density parity check (LDPC) and good quasi-cyclic LDPC error correcting codes in polynomial time.

Before we delve into all the details of this thesis, we provide in Chapter 2 a mostly self-contained proof of a simplification of a core technical tool that will be refined throughout this thesis. This proof illustrates a lot of the ideas that will show up in the following chapters.

In Chapter 3 we describe some of the background of this thesis, including techniques and tools that will be used repeatedly.

Chapter 4 is based on the results of [MOP20a]. We show how to construct near-Ramanujan graphs in a probabillistically strongly explicit way. We also develop some of the main ideas that are generalized and used in the following chapters.

Chapter 6 is based on the results of [MOP20b]. We precisely determined the SDP value of large random instances of certain kinds of constraint satisfaction problems. To establish this, we analyze the spectral expansion of a distribution of graphs that generalizes uniformly random regular graphs.

Chapter 5 is based on the results of [Par21]. We describe a new method to remove short cycles on regular graphs while maintaining spectral bounds as long as the graphs have certain combinatorial properties. Using this method we were able to show two results involving high girth spectral expander graphs.

Chapter 7 is based on the results of [JMO$^+$22]. We study a generalization of lifts based on groups and we describe explicit constructions of expanders obtained via abelian lifts. We use

this to obtain explicit quantum lifted product codes of almost linear distance. As a corollary, we also obtain good quasi-cyclic LDPC codes with any circulant size up to nearly linear.

Finally in Chapter 8 we discuss some open problems related to this thesis and present some closing remarks.

Chapter 2

A Self-Contained Proof of the Main Technical Tool

In this chapter we will show a mostly self-contained proof of a simplification of the main tool we refine and repeatedly use throughout this thesis. In the process, we will also discuss all the background on the tools and ideas that went into the proof. It is perhaps the main technical contribution of this thesis and so to a reader interested in some of the technical aspects of this thesis but that does not want to go through the whole document, we recommend reading just this chapter.

Before we keep going, let us set the stage. Our general goal is to build expanding graphs so suppose we are given an integer n, an integer d and a positive constant real ε and we want to build a graph G with around n vertices, say $\Theta(n)$, that is d-regular and is also ε-near-Ramanujan, that is, such that $\lambda(G) \leqslant 2\sqrt{d-1} + \varepsilon$. By build we mean we want a weakly explicit construction, which recall means we want a polynomial time (in n) algorithm that produces such a graph.

Our strategy to do the above can be summarized as follows. Let us suppose that we have an efficient tool to increase the size of a graph while keeping its regularity and spectral expansion. Then, we can start by producing a small base graph that is a good expander and then we can apply this tool multiple times to obtain a graph with $\Theta(n)$ vertices. For example, we could start with the complete d-regular graph [1], which is a graph with $d+1$ vertices which has optimal spectral expansion.

The magical operation that grows the size of a graph we will use is called a *2-lift*. To 2-lift a graph $G = (V, E)$ we start by creating a single copy of each vertex in V. Then, for each edge $e = \{u, v\} \in E$ we will place two edges in the lifted graph. Let u', v' be the copies of u, v, we can either "parallel connect" them by placing the edges $u \sim v$ and $u' \sim v'$, or we "cross connect" them by placing the edges $u \sim v'$ and $u' \sim v$.

A uniformly random 2-lift of G is a graph where the parallel/cross choice for each edge is uniformly random and independent. We will see soon that uniformly random 2-lifts of "good" graphs are really good expanders, but first we need one more definition to be able to specify what "good" means in this context.

[1] As we will see, we cannot actually use the complete d-regular graph as our base graph, because we require some extra properties it does not have. We will discuss this further later in this thesis.

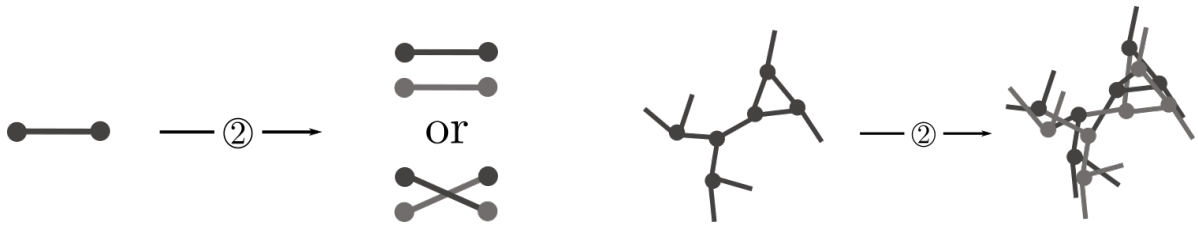

Figure 2.1: Illustrations of the 2-lift operation

A *bicycle-free* graph is a graph that contains at most one cycle. A graph $G = (V, E)$ is *bicycle-free at radius* r if for all vertices $v \in V$, the neighborhood of radius r around v, meaning the set of vertices at a distance of at most r from v, is bicycle-free.

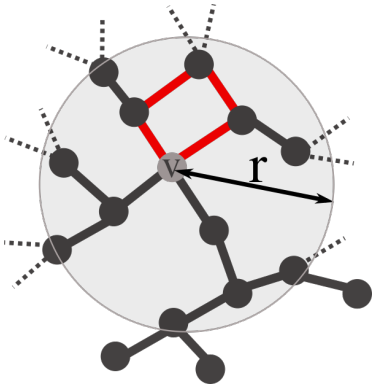

Figure 2.2: A bicycle-free graph at radius r

Notice that this definition can be interpreted as "almost" having girth $2r$ since having girth $2r$ is equivalent to having the property that for all vertices $v \in V$ the neighborhood of radius r around v is cycle-free.

2.1 Statement of the theorem

We can finally state the theorem we will prove in this section.

Theorem 2.1.1. *Let G_0 be an n-vertex d-regular graph such that:*

- *G_0 is ε-near Ramanujan.*
- *G_0 is bicycle-free at radius $r = \Theta(\log_{d-1} n)$.*

Let G be a uniformly random 2-lift of G_0. Then,

$$\lambda(G) \leqslant \frac{5}{2}\sqrt{d-1} \cdot (1+\varepsilon)$$

with probability at least $1 - \exp(-n^{\Theta_{\varepsilon,d}(1)})$.

In Chapter 4 we generalize and improve on this theorem by reducing the $\frac{5}{2}$ to the optimal constant 2 and by derandomizing it. The generalized proof is more technical and has some extra

steps, but it is pretty similar to the one we describe here. As we go to the proof in this chapter we will point out what steps of the proof require some extra ingredients to obtain the generalized proof. Additionally, in Chapter 7 we further generalize this proof to a larger class of lifts called abelian lifts.

Why do we assume bicycle-freeness instead of "cycle-freeness" or "tricycle-freeness"? When applying this core tool we will need to start with a base graph that satisfies all the assumptions. It turns out that constructing regular graphs that are cycle-free at high radius is really hard, however a uniformly random d-regular graph is bicycle-free at radius $\Theta(\log_{d-1} n)$ with high probability. Interestingly, it will not be too hard to see that our proof still applies (with some small modifications) if the base graph is "constant-cycle-free" at high radius, i.e. if there is a value $r = \Theta(\log_{d-1} n)$ for which all neighborhoods of radius r have a number of cycles bounded by some constant. We will justify the spectral assumption of this theorem as we go through the proof.

We divide the exposition into three parts. We first massage the theorem statement by applying some known facts about 2-lifts and bounding eigenvalues of matrices. Once we obtain a more managable statement, its proof will have two parts: a random part and a deterministic part. In the random part we use the probabilistic properties of uniformly random 2-lifts to reduce the proof to solving a deterministic combinatorial problem. In the deterministic part we solve this combinatorial problem.

2.2 Setting up the proof

The spectrum of 2-lifts Let us start by describing an equivalent definition of 2-lefts of a graph $G = (V, E)$. Consider an edge-signing $w : E \to \{\pm 1\}$ of G. This edge-signing uniquely defines a 2-lift $\overline{G} = (\overline{V}, \overline{E})$ of G, which we can describe in the following way:

$$\overline{V} = V \times \{\pm 1\}, \qquad \overline{E} = \Big\{ \{(u, \sigma), (v, \sigma \cdot w(u, v))\} : (u, v) \in E, \sigma \in \{\pm 1\} \Big\}.$$

This shows there is a bijection between edge-signs and 2-lifts. For example, we can redefine a uniformly random 2-lift as a 2-lift given by a uniformly random edge-signing, that is an edge-signing where the sign of each edge is ± 1 with probability $1/2$ and independent of other signs. This bijection is very useful because it helps us characterize the spectrum of $\overline{G}$. Let $\mathrm{Spec}(M)$ be the set of eigenvalues of a matrix M. Let A be the adjacency matrix of G, $\overline{A}$ the adjacency matrix of the lifted graph $\overline{G}$ and finally A_w the adjacency matrix of the graph G signed by w. Formally, $(A_w)_{uv} = w(u, v)$ for all edges $\{u, v\} \in E$ and 0 otherwise. Notice that A and A_w are n by n matrices, but $\overline{A}$ is a $2n$ by $2n$ matrix. We can now show the following lemma:

Lemma 2.2.1. *We have:*

$$\mathrm{Spec}(\overline{A}) = \mathrm{Spec}(A) \cup \mathrm{Spec}(A_w).$$

Proof. Let A_+ be the positive adjacency matrix, meaning the adjacency matrix that only contains the edges in E with positive w sign, and similarly define A_-. Formally, $(A_\pm)_{uv} = 1$ if $\{u, v\} \in E$ and $w(u, v) = \pm$. Note that $A = A_+ + A_-$ and $A_w = A_+ - A_-$. We can write the adjacency

matrix of $\overline{A}$ in the following block form, where we index first the original vertices and then the copies:

$$\overline{A} = \begin{bmatrix} A_+ & A_- \\ A_- & A_+ \end{bmatrix}.$$

Suppose ν is an eigenvector of A with eigenvalue λ, then let $\overline{\nu} = (\nu \ \nu)$ and observe that $\overline{A}\overline{\nu} = (A_+\nu + A_-\nu, A_+\nu + A_-\nu) = (A\nu, A\nu) = (\lambda\nu, \lambda\nu) = \lambda\overline{\nu}$, so $\overline{\nu}$ is an eigenvector of $\overline{A}$ with eigenvalue λ.

Now suppose ν is an eigenvector of A_w with eigenvalue λ, then let $\overline{\nu} = (\nu \ -\nu)$ and observe that $\overline{A}\overline{\nu} = (A_+\nu - A_-\nu, -A_+\nu + A_-\nu) = (A_w\nu, -A_w\nu) = (\lambda\nu, -\lambda\nu) = \lambda\overline{\nu}$, so $\overline{\nu}$ is an eigenvector of $\overline{A}$ with eigenvalue λ.

Notice that all of these vectors are orthogonal and there are $2n$ of them, so this completely describes the spectrum of $\overline{A}$. □

Notice that this lemma tells us that the eigenvalues of a 2-lift of a graph are the original eigenvalues as well as the eigenvalues given by the signed adjacency matrix, using the edge-signing that corresponds to the 2-lift. Hence, if we are 2-lifting a base graph that is a good expander we only need to worry about the "new" eigenvalues induced by the signed adjacency matrix. Using this idea, we can simplify the theorem statement we want to prove. Let $\rho(M)$ [2] be the largest eigenvalue of M.

Theorem 2.2.2 (Implies Theorem 2.1.1)**.** *Let G_0 be an n-vertex d-regular graph such that:*

- *G_0 is bicycle-free at radius $r = \Theta(\log_{d-1} n)$.*

Let w be a uniformly random edge-signing of G_0. Then,

$$\rho(A_w) \leqslant \frac{5}{2}\sqrt{d-1} \cdot (1+\varepsilon)$$

with probability at least $1 - \exp(-n^{\Theta_{\varepsilon,d}(1)})$.

We highlighted the differences from the original statement. Notice that we also dropped the requirement that G_0 is ε-near-Ramanujan. This does not mean we do not need this for the original theorem, but observe that since a 2-lift keeps the eigenvalues of the base graph, if we want our lifted graph to be expanding we need the base graph to be ε-near-Ramanujan [3]. But in this rewritten statement we only analyze the "new" eigenvalues given by the signed graph. Thus, the correctness of this statement implies the correctness of the original statement.

Bounding eigenvalues Our task now is to bound the largest eigenvalue of a certain matrix A_w. There are many ways of analyzing the largest eigenvalue of symmetric real matrices, but we are going to use one that is known as the *trace method*, also known as the Füredi-Komlós Trace Method [FK81]. Before we do so let us recall two important properties of the eigenvalues of symmetric matrices.

[2] Later in this thesis we use ρ to represent the *spectral radius* of an operator, but for symmetric real matrices this is just the largest eigenvalue so we simplified the notation here.

[3] Actually, in the version of this result that we are proving we only need $\lambda(G_0) \leqslant \frac{5}{2}\sqrt{d-1} + \varepsilon$, but we kept the ε-near-Ramanujan since we will need it in the later chapters when we look at the more general cases.

Let M be a $n \times n$ symmetric real matrix and denote its eigenvalues by $\lambda_1 \geqslant \ldots \geqslant \lambda_n$. It is easy to show (take the trace of the eigendecomposition of M) that the trace of M is equal to the sum of the eigenvalues, that is $\operatorname{tr}(M) = \sum_i \lambda_i$, this is known as the *trace identity*. Now, for a non-negative integer k consider the kth power of M. The set of eigenvalues of M^k (which we denote by $\operatorname{Spec}(M^k)$) is exactly the set $\{\lambda_1^k, \ldots, \lambda_n^k\}$ (the proof a simple induction argument), this is known as the *power identity*.

We can now apply these two identities to A_w. Let k be a positive even integer, then we have the following:

$$\rho(A_w^k) = \lambda_1^k \leqslant \sum_i \lambda_i^k = \operatorname{tr}(A_w^k).$$

Using this fact we can bound $\rho(A_w)$ by bounding $(\operatorname{tr}(A_w^k))^{1/k}$ for a choice of even positive k. To see how powerful this statement is, consider the kth power of the adjacency matrix of G, so A^k. It is easy to see that this matrix represents all the walks of length k, that is the entry $(A^k)_{uv}$ for $u, v \in V$ is equal to the number of walks of length k that start in u and end in v (this is another simple induction argument). Similarly, A_w^k represents w-signed walks of length k, or more formally $(A_w^k)_{uv} = \sum_\gamma \prod_i w(e_i)$, where $e_i \in E$ and $\gamma = (e_1, \ldots, e_k)$ is a walk of length k such that $e_1 = (u, \cdot)$ and $e_k = (\cdot, v)$. Finally the trace of A_w^k is the sum of the signed *closed* walks of length k, i.e. walks that start and end at the same vertex.

So to bound the largest eigenvalue of A_w all we have to do is to know the sum of the closed w-signed walks of a chosen length in G. This is a purely combinatorial problem and much easier to reason about than the algebraic notions of eigenvalues. This is the power of the trace method.

Counting closed walks We have made great strides in our search for an eigenvalue bound, however it turns out that counting closed walks in graphs is still a hard task. To address this we can try to instead count *non-backtracking* walks. These are walks that never backtrack, that is, we never take the same edge twice in a row (but we might repeat it later in the walk).

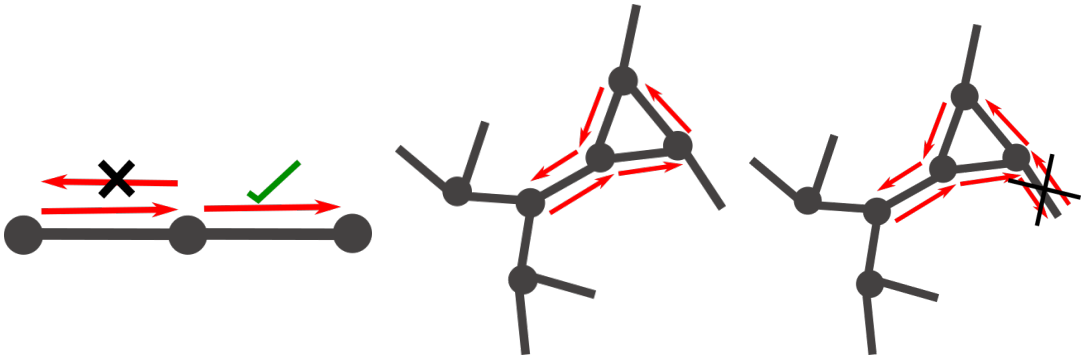

Figure 2.3: An illustration of non-backtrackness and two example walks: a non-backtracking walk and a walk that backtracks.

Why would one count non-backtracking walks instead of the usual walks? To illustrate how easy counting these walks is, consider a tree graph and suppose we want to count how many non-backtracking walks are there between two given vertices. It is easy to see that there is only

one unique walk. However, the number of (possibly backtracking) walks between two vertices on a tree is much harder to count.

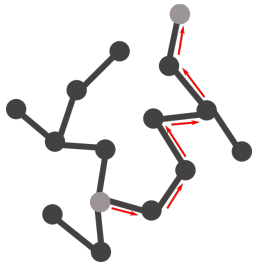

Figure 2.4: A non-backtracking walk on a tree

We now know that (at least intuitively) counting non-backtracking walks is easier, but our original task is to count normal walks. As it turns out, we can relate the number of closed walks to the number of non-backtracking walks. To do so, we will need to use a matrix analogous to the adjacency matrix called the *non-backtracking matrix*. For ease of reading, we will delay formally defining this until the later chapters and instead we will give an informal definition.

The non-backtracking matrix of the graph G, which we shall denote by B, is a $2|E|$ by $2|E|$ matrix indexed by the directed edges of G. The directed edges of G are obtained by taking each undirected edge $e = \{u, v\} \in E$ and replacing it with two directed edges in each direction, namely (u, v) and (v, u). We define B as the matrix that has the property that for any positive integer k and pair of directed edges $\vec{e}$ and $\vec{e'}$, the entry $(B^k)_{\vec{e}\vec{e'}}$ is equal to the number of length k non-backtracking walks that take $\vec{e}$ as the first step and $\vec{e'}$ as the final step. Similarly, we define B_w as the w-signed non-backtracking matrix, where we consider w-signed walks instead. Slightly more formally we have:

$$(B^k)_{\vec{e}\vec{e'}} = \sum_{\vec{e}=\vec{e_1},\ldots,\vec{e_k}=\vec{e'} \text{ non-backtracking}} \prod_{i=2}^{k} w(e_i).$$

Notice that the product term goes from 2 to k, this is due to a technical reason related to the definition of B that is not very important. In the next few chapters we will address this in more detail when we define B more formally.

Perhaps surprisingly, the *Ihara–Bass formula* gives us a map to translate between eigenvalues of A (or A_w) and the eigenvalues of B (or B_w). Formally it says the following:

Theorem 2.2.3. *(Ihara–Bass formula.) Let $\lambda \neq 0, \pm 1$ be a number such that $\lambda + (d-1)/\lambda$ is an eigenvalue of A (respectively A_w). Then λ is an eigenvalue of B (respectively B_w).*

This is the only statement we will not prove in this section. We will further discuss it and show a short proof of this in Chapter 3. By applying this formula it is easy to conclude the following corollary:

Corollary 2.2.4. *If A has an eigenvalue of magnitude $\frac{5}{2}\sqrt{d-1} + \varepsilon$ (for $\varepsilon \geqslant 0$) then B has an eigenvalue of magnitude $2\sqrt{d-1} + \Theta_d(\varepsilon)$.*

After all of this work, we can apply this to our original question and rewrite it one last time into the following:

Theorem 2.2.5 (Implies Theorem 2.1.1)**.** *Let G_0 be an n-vertex d-regular graph such that:*

- *G_0 is bicycle-free at radius $r = \Theta(\log_{d-1} n)$.*

Let w be a uniformly random edge-signing of G_0. Then,

$$\rho(B_w) \leqslant 2\sqrt{d-1} \cdot (1+\varepsilon)$$

with probability at least $1 - \exp(-n^{\Theta_{\varepsilon,d}(1)})$.

Before we go on, it is important to note that to apply the trace method to B_w we have to be a bit more careful than before because B_w is not a symmetric matrix and it might have complex eigenvalues. However, we can use the following version of the trace method. Let M be a square real matrix.

$$\operatorname{tr}(M^{2k}) = \operatorname{tr}(M^k (M^T)^k) = \|M^k\|_F^2 = \sum_i |\lambda_i^k|^2 \geqslant \rho(M)^{2k},$$

where $\|M\|_F = \sqrt{\sum_{i,j} M_{ij}^2}$ is the Frobenius norm of M.

We will now focus on proving the above theorem statement that implies Theorem 2.1.1.

2.3 The random part

After the discussion of the previous section it should be no surprise that our first step will be to apply the trace method to B_w. So let k be a positive integer which we will pick later. We have

$$\begin{aligned}
\rho(B_w)^{2k} &\leqslant \operatorname{tr}(B_w^k (B_w^T)^k) \\
&= \sum_{\vec{e_0},\cdots,\vec{e_{2k}} = \vec{e_0} \in \vec{E_0}} (B_w)_{\vec{e_0},\vec{e_1}} \cdots (B_w)_{\vec{e_{k-1}},\vec{e_k}} (B_w)^T_{\vec{e_k},\vec{e_{k+1}}} \cdots (B_w)^T_{\vec{e_{2k-1}},\vec{e_{2k}}} \\
&= \sum_{\vec{e_0},\cdots,\vec{e_{2k}} = \vec{e_0} \in \vec{E_0}} (B_w)_{\vec{e_0},\vec{e_1}} \cdots (B_w)_{\vec{e_{k-1}},\vec{e_k}} (B_w)_{\vec{e_{k+1}},\vec{e_k}} \cdots (B_w)_{\vec{e_{2k}},\vec{e_{2k-1}}},
\end{aligned}$$

where $\vec{E_0}$ is the set of directed edges of G_0.

Careful observation of the formula above shows that we need to look at walks that non-backtrack for k steps, then they take the reverse of the kth step, followed by k more steps. To capture this idea we make some definitions.

Definition 2.3.1 (Hikes)**.** For $\ell \in \mathbb{N}$, we define an *ℓ-hike* $\mathcal{H}$ to be a closed walk in G_0 of exactly 2ℓ steps (directed edges) which is non-backtracking except possibly between the ℓth and $(\ell+1)$th step. Given an edge-signing $w : E \to \{\pm 1\}$ we write $w(\mathcal{H})$ for the product of the edge-signs that $\mathcal{H}$ traverses, counted with multiplicity.

Finally, we call a $(\ell+1)$-hike *special* if the $(\ell+1)$th step is the reverse of its $(\ell+2)$th step, and the last step is the reverse of the first step.

So we now simplify the above into:

$$\rho(B_w)^{2k} \leqslant \sum_{\text{special } (k+1)-\text{hikes } \mathcal{H} \text{ in } G_0} w(\mathcal{H}).$$

Eventually we want to bound this random variable with high probability. Let $T = \mathrm{tr}(B_w^k (B_w^T)^k)$, which is a random non-negative variable (since it is equivalent to $\|(B_w)^k\|_F^2$) and an upper bound to $\rho(B_w^k)$. To bound this random variable we will use a very standard technique where we first compute the expected value of T and then we apply Markov's inequality. So we will spend most of our time upper bounding the expectation of this random variable. Using linearity of expectation and the above we can write (we write w in the subscript below to highlight that the expectation is over the randomness of w):

$$\mathbf{E}_w[T] = \sum_{\text{special } (k+1)-\text{hikes } \mathcal{H} \text{ in } G_0} \mathbf{E}_w[w(\mathcal{H})].$$

Now comes the crucial observation that makes this computation possible. Consider some special k-hike $\mathcal{H}$ and suppose there is some (undirected) edge e that is traversed (in either direction) by $\mathcal{H}$ exactly once. When computing $\mathbf{E}_w[w(\mathcal{H})]$ notice that the contribution of $w(e)$ is independent of the rest of the walk, so we can write that expectation as $\mathbf{E}_w[w(e) \cdot w(\mathcal{H}')] = \mathbf{E}_w[w(e)] \cdot \mathbf{E}_w[w(\mathcal{H}')]$, where $\mathcal{H}'$ is a walk that does not traverse e. But now observe that $\mathbf{E}_w[w(e)] = \mathbf{E}_{x\sim\{\pm 1\}}[x] = 0$. We conclude that actually $\mathcal{H}$ does not contribute anything to the expectation of T. To account for this we make one more definition.

Definition 2.3.2 (Singleton-free hikes)**.** We call a hike *singleton-free* if each undirected edge traversed by $\mathcal{H}$ is traversed at least twice.

Note that for singleton-free hikes $\mathcal{H}$ we have that $\mathbf{E}_w[w(\mathcal{H})] \leqslant 1$, since this is a product of ± 1. And so we can write:

$$\mathbf{E}_w[T] \leqslant |\{\text{special, singleton-free } (k+1) - \text{hikes } \mathcal{H} \text{ in } G_0\}|.$$

Finally, every singleton-free special $(k+1)$-hike $\mathcal{H}$ can be formed from an singleton-free $(k-1)$-hike $\mathcal{H}'$ by: (i) attaching a step and its reverse to the beginning/end of $\mathcal{H}$; (ii) attaching a step and its reverse to the midpoint of $\mathcal{H}$. As there are at most $(d-1)^2 \leqslant d^2$ choices for how to perform (i) and (ii), this shows the following proposition:

Proposition 2.3.3.

$$\mathbf{E}_w[T] \leqslant d^2 |\{\textit{singleton-free } (k-1) - \textit{hikes } \mathcal{H} \textit{ in } G_0\}|.$$

Detour: Pseudorandomness

To obtain the derandomized version of Theorem 2.1.1 this step will be slightly different. Instead of using perfectly uniformly random edge-signings, we will use a standard derandomization tool called (δ, k)-*wise uniform bits*. These define a distribution over $\{\pm 1\}^n$ that we can sample from very efficiently and that is δ-close to the uniform distribution when restricted to k coordinates.

Using these, the expectation $\mathbf{E}_w[w(\mathcal{H})]$ will not be 0, but instead it will be small enough to not affect the overall bound.

The work we just did allowed us to go from counting hikes to counting singleton-free hikes. This is a seemingly innocent, but crucial step in our proof of the original statement. Let us forget about hikes for just a second and consider counting non-backtracking walks of length $2k$ in G_0, a d-regular graph. A trivial upper bound on the number of such walks is $(d-1)^{2k}$.

Recall that to apply the trace method we are trying to bound $\rho(B_w)^{2k}$ in high probability, so if we use the bound we just discussed on the number of non-backtracking walks of length $2k$ and then we take $2k$th roots we get that $\rho(B_w) \leqslant d-1$. This is a trivial bound, since after applying the Ihara—Bass formula we get that $\rho(A_w) \leqslant d$, which is the trivial upper bound that gives no expansion.

Now, consider "singleton-free" non-backtracking walks of length $2k$. Since we have to traverse each undirected edge at least twice, intuitively we are only really taking k unique new steps. Consider the infinite d-regular tree like the one in Figure 2.5 and suppose we want to count how many non-backtracking walks of length $2k$ where we can backtrack on step k there are starting at a given vertex. The first k steps have $(d-1)^k$ possibilities, but the remaining k steps are fully determined since we need to traverse each edge twice, so in total there are $(d-1)^k$ possible walks. When we factor this into the trace method calculation we get that $\rho(B_w) \leqslant \sqrt{d-1}$, which when applying the Ihara—Bass formula gives us the Ramanujan bound of $\rho(A_w) \leqslant 2\sqrt{d-1}$.

The finite case is not as nice as the infinite case, but the intuition is the same: if we only take about k unique steps out of the total $2k$, then asymptotically the total number of walks grows like $(d-1)^k$ and the trace method gives us a bound of close to $\sqrt{d-1}$. In this chapter we will not obtain this optimal bound, but something slightly worse, that is reserved for the later chapters.

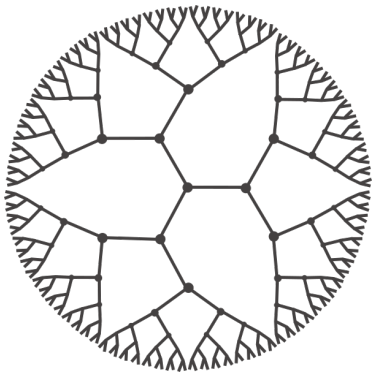

Figure 2.5: An infinite 3-regular tree.

We have reduced our problem to the problem of counting singleton-free $(k-1)$-hikes in G_0. This is a purely deterministic combinatorial problem that we will tackle in the next section.

2.4 The deterministic part

Counting walks in our graph G_0 is now our only focus. We will use an *encoding argument* to do so, but first let us briefly see what that means.

Suppose you want to count how many objects are in some set S. Consider some injective function $C : S \to \Sigma^\star$, where Σ is a set of symbols (e.g. binary digits, so $\Sigma = \{0, 1\}$), meaning

we have some way to encode each element of S into a string, such that any two elements have a different encoding. Another way of saying this is to say that C is reversible, given some encoded string c there is one unique element s such that $C(s) = c$. Now, suppose we can show that for all elements $s \in S$ the length of its encoding $|C(s)|$ is at most some value ℓ. Then, we conclude that $|S|$ is at most $|\Sigma|^\ell$. This is the basic idea of what we will do in this section.

Now, let us go back to the problem of counting singleton-free hikes. Let $\mathcal{H}$ be some singleton-free $(k-1)$-hike in G_0. We will encode $\mathcal{H}$ in two steps: first we will encode what we call the *hike graph*, which we will define shortly; then, given the hike graph, we encode $\mathcal{H}$.

2.4.1 Encoding the hike graph

The *hike graph* of $\mathcal{H}$, written as $G_{\mathcal{H}} = (V(\mathcal{H}), E(\mathcal{H}))$, is the undirected graph induced by $\mathcal{H}$, so the union of the undirected edges that $\mathcal{H}$ traverses along with their endpoints. We want to encode $G_{\mathcal{H}}$, so we want some reversible function that maps $G_{\mathcal{H}}$ into a string.

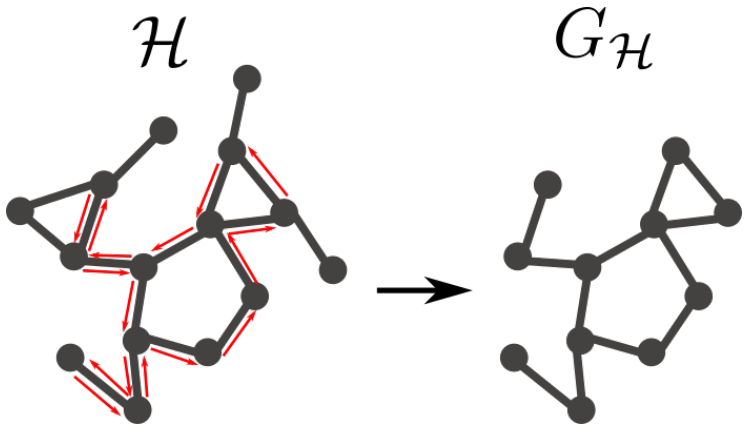

Figure 2.6: The hike graph

We will do this by encoding the Depth-First Search (DFS) traversal of it from a given start vertex. So let us suppose we pick some arbitrary vertex $v \in V(\mathcal{H})$ from which we start a DFS traversal. As in the typical DFS algorithm, we recursively process each vertex by first checking whether we have visited it. If we have not, we iterate through all of its neighbors but its "parent", the vertex from which we came (except for the very first vertex, since it is the first one we are processing). We process each neighbor recursively one by one. If we have visited the current vertex we simply return/backtrack to the previous vertex. Each time we make a recursive call to process a new vertex we call it a *recursive step* and each time we finish processing a vertex, either because we looked at all its neighbors or because we had visited it before, we call it a *backtracking step*.

As we traverse the graph, we will write down a "trace log" by keeping track of two types of data before every step - (1) Is this step recursive or backtracking (2) If it is a recursive step, then which neighbor do we recurse to. For the type of data (2) we can assume that for each vertex $v \in V_0$ we pick some arbitrary order of its d neighbors, and we use this number as the information of which neighbor we recurse to. Note that by writing down this information we can perform the DFS in "reverse" and reconstruct the original graph.

How do we encode this trace log? We have to encode three different things: the initial vertex; for each step, whether it is recursive or backtracking; for each recursive step, the index of the neighbor we recursed to. Note that there are $2|E(\mathcal{H})|$ total steps, two per edge, since each edge produces a recursive and a backtracking steps. Also, there are $|E(\mathcal{H})|$ recursive steps for the same reason. Finally, there is obviously only one initial vertex.

This means we encode the initial vertex by keeping track of a number $v \in [n]$ (recall that $n = |V_0|$). We encode the type of each step by writing a sequence of binary symbols $\sigma \in \{\mathtt{R}, \mathtt{B}\}^{2|E(\mathcal{H})|}$. We encode the neighbor index of each recursive step by writing a sequence of numbers $(d_1, \ldots, d_{|E(\mathcal{H})|}) \in [d-1]^{|E(\mathcal{H})|}$ [4].

In the figure below we have an example of a hike graph and the DFS tree that is produced by traversing it: the straight edges represent vertices that we are processing for the first time and the round edges repeated visits.

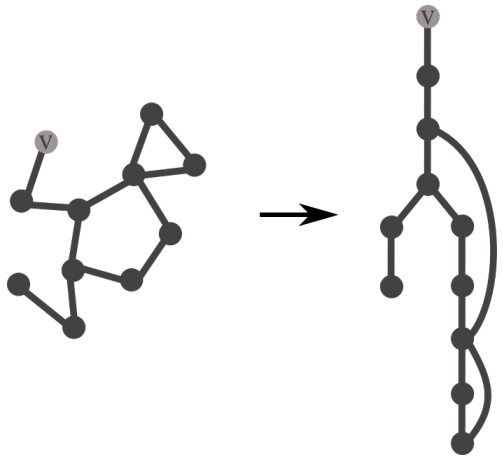

Figure 2.7: Traversing a graph using a DFS

The first few symbols of the encoding of this graph would be something like:

v

$\mathtt{R}\ d_1$

$\mathtt{R}\ d_2$

$\mathtt{R}\ d_3$

$\mathtt{R}\ d_4$

$\mathtt{R}\ d_5$

$\mathtt{B}$

$\mathtt{B}$

$\mathtt{R}\ d_6$

$\mathtt{R}\ d_7$

[4] We are cheating slightly here since the first recursive step can be one of d neighbors, but this is a minor technical detail that will not affect the final calculation.

We can now use this encoding to show the following lemma:

Lemma 2.4.1. *The number of connected subgraphs of G_0 having at most k edges is at most*

$$2n \cdot (d-1)^k \cdot 2^{2k}.$$

Proof. Let us first count the number of such subgraphs with exactly m edges. Using the DFS encoding recall that we need to encode:

- $v \in [n]$.
- $(d_1, \ldots, d_m) \in [d-1]^m$.
- $\sigma \in \{\mathtt{R}, \mathtt{B}\}^{2m}$.

And so we conclude that an upper bound on the number of subgraphs with exactly m edges is $n \cdot (d-1)^m \cdot 2^{2m}$. To get the result we desire we have to sum over all integer $1 \leqslant m \leqslant k$, since this is a geometric sum we get the lemma. □

Note that we only need to count hike graphs with at most k edges because we are looking at singleton-free hikes that take at most $2k$ steps, so half of the steps we take have to traverse edges we have seen before, which does not add to the number of edges in the hike graph. As we discussed before, this is crucial to obtain a non-trivial bound.

Detour: Improving the spectral bound through a better encoding

It is not clear why yet, but this is where we need to improve our encoding in order to obtain a bound of $2\sqrt{d-1}$ instead of the $\frac{5}{2}\sqrt{d-1}$ we are working towards in this chapter.
We did not use any information about hike graphs other than the fact that they are subgraphs of d-regular graphs with at most k edges. In fact, these graphs have a lot of structure that we can use to make our encoding more efficient and hence obtain a better upper bound. The main insight, which is not obvious without first making a few observations, is that hike graphs are really sparse and will often have mostly vertices with degree 2. This means when we backtrack we often backtrack multiple times in a row, so we can encode that by writing down how many times we backtrack in a row instead of a long string of Bs.

2.4.2 Encoding the walk

We have not yet used the fact that the base graph G_0 is bicycle-free at radius r. This will be a crucial ingredient in this second part of the encoding as we finally encode our hike $\mathcal{H}$ assuming it has a given graph H as its hike graph.

To simplify our job, let us first partition $\mathcal{H}$ into blocks of r consecutive steps (for simplicity, the reader can assume $|\mathcal{H}| = 2(k-1)$ is a multiple of r, but this assumption is not needed if we are a bit more careful and suppose that there might be one block with less than r steps). This results in $2k/r$ blocks. Since $\mathcal{H}$ is walking on a graph that is bicycle-free at radius r, this means that each block induces a bicycle-free graph. We encode each block separately and then join the resulting encodings together.

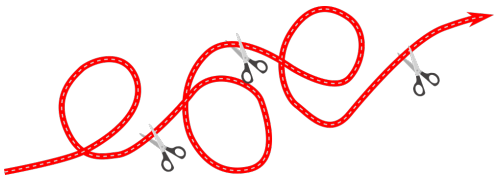

Figure 2.8: Partitioning $\mathcal{H}$ into $2k/r$ blocks of k steps.

So we are trying to encode a non-backtracking walk of length r on a bicycle-free graph. If the graph we are walking on were "cycle-free" (i.e. a tree) then to encode a walk we only need to record the first and last vertex of the walk, since there is only one possible non-backtracking walk between any two vertices. The bicycle-free case is not that simple, but fortunately it is not much harder. Notice that if we know the first and final vertices u, v of the walk then all possible non-backtracking walks do the following: walk from u to the only cycle in the hike graph; traverse this cycle in one of two possible directions some positive number c times; leave the cycle and walk to v. We can also walk directly from u to v, which we can assume is equivalent to traversing the cycle $c = 0$ times.

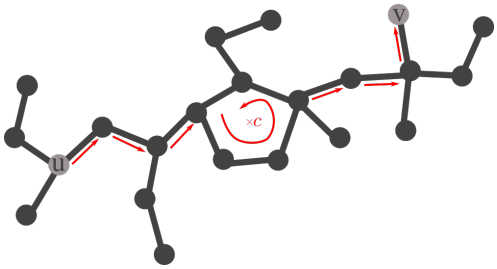

Figure 2.9: All possible walks between u and v in a bicycle-free graph.

We conclude that to encode one block we need to keep track of where we start the block walk, which is an integer $u \in [|V(\mathcal{H})|]$, where we end the block walk, which is an integer $v \in [|V(\mathcal{H})|]$, and some information about the walking on the cycle. We need to keep track of how many times (at most $\lfloor r/2 \rfloor$) we traverse the cycle and which direction we did so (positive or negative), so we can keep track of an integer $c \in \{0, \ldots, \pm\lfloor r/2 \rfloor\}$.

We can now use this encoding to show the following lemma:

Lemma 2.4.2. *Given an r-bicycle-free hike graph H, the number of singleton-free $(k-1)$-hikes that have H as their hike graph is at most*

$$(r|V(H)|)^{2k/r}.$$

Proof. We use the encoding above and partition a hike into $2k/r$ bicycle-free blocks. Notice that the final vertex of a block is the first vertex of the next block, so we need to encode:

- $(v_1, \ldots, v_{2k/r}) \in [|V(H)|]^{2k/r}$ final vertices.
- $(c_1, \ldots, c_{2k/r}) \in \{0, \ldots, \pm\lfloor r/2 \rfloor\}^{2k/r}$ cycle information.

Using this encoding we obtain the desired result. □

2.4.3 Full encoding

We now put everything together to obtain the final bound on singleton-free hikes.

Lemma 2.4.3.

$$|\{\textit{singleton-free } (k-1)\textit{-hikes } \mathcal{H} \textit{ in } G_0\}|$$

is at most

$$\left(2^{\gamma}\sqrt{d-1}\right)^{2k},$$

where $\gamma = 1 + \frac{\log(2nrk)}{2k} + \frac{\log(rk)}{r}$

Proof. Combining the two encoding lemmas and using the fact that any hike graph has at most k vertices, we obtain the following bound:

$$\begin{aligned}|\{\text{singleton-free } (k-1)\text{-hikes } \mathcal{H} \text{ in } G_0\}| &\leqslant 2n \cdot (d-1)^k \cdot 2^{2k} \cdot (rk)^{2k/r} \\ &\leqslant (2nrk) \cdot (rk)^{2k/r} \cdot (2\sqrt{d-1})^{2k} \\ &\leqslant \left((2nrk)^{1/2k} \cdot (rk)^{1/r} \cdot 2\sqrt{d-1}\right)^{2k}\end{aligned}$$

We take the logarithm of both sides and we obtain the lemma. □

We will work in a regime where $k \gg \log n$ so the term $\frac{\log(2nrk)}{2k}$ is $o(1)$.

2.5 The final countdown

After all the work we did over the last two sections we managed to prove that:

$$\mathbf{E}[\rho(T)] \leqslant d^2(2^{\gamma}\sqrt{d-1})^{2k},$$

where $\gamma = 1 + o(1) + \frac{\log(rk)}{r}$ and T is an upper bound on $\rho(B_w)^{2k}$. So now it is finally time to apply Markov's inequality. Let $\eta \in (0,1)$ be a number to be picked later. Markov gives us that with probability at least $1-\eta$

$$\rho(B_w)^{2k} \leqslant (d^2/\eta)(2^{\gamma}\sqrt{d-1})^{2k}.$$

We can take the $2k$th roots of the above, apply Bernoulli's inequality[5], rearrange and obtain

$$\rho(B_w) \leqslant 2^{\gamma'}2\sqrt{d-1} \leqslant 2\sqrt{d-1} \cdot (1+\gamma'),$$

where $\gamma' = \frac{\log(d^2/\eta)}{2k} + o(1) + \frac{\log(rk)}{r}$.

To obtain our desired result we need to pick our parameters k and η such that $\gamma' \leqslant \varepsilon$. There are multiple possible choices of parameters that lead to interesting results. We will discuss this in the later chapters, but here we just show one possible choice that gives us the desired theorem statement. Let us set $\varepsilon_1 = \frac{\log(d^2/\eta)}{2k}$ and $\varepsilon_2 = \frac{\log(rk)}{r}$ and try to choose parameters such that $\varepsilon_i \leqslant \varepsilon/2$. Here is a list of our parameters and their choices:

[5] $(1+x)^r \leqslant 1 + rx$ positive x and $0 \leqslant r \leqslant 1$

- $r = c \log_{d-1} n$, this is given by the theorem statement.
- $k = \exp(\varepsilon r/2) = n^{\delta}$, where $\delta = O_{\varepsilon,d}(1)$, which makes $\varepsilon_2 \leqslant \varepsilon/2$.
- $\eta = \exp(-n^{\delta}\varepsilon/2)$, which makes $\varepsilon_1 \leqslant \varepsilon/2$.

Finally, we get that with probability $\eta = 1 - \exp(-n^{\Theta_{\varepsilon,d}(1)})$

$$\rho(B_w) \leqslant 2\sqrt{d-1} \cdot (1+\varepsilon).$$

This concludes the proof of the main result of this chapter.

Chapter 3

Background

In this chapter we will introduce several tools and techniques that will be useful through out this thesis. We reserve this chapter for the background that will be important for the whole thesis, and thus some of the chapters to come will include an extra background section to introduce topics that only pertain to that chapter.

3.1 Graphs and linear algebra

As in the previous chapter, consider an undirected n-vertex multigraph $G = (V(G), E(G))$. A *walk* on G is a sequence of vertices $(v_1, v_2, \ldots, v_n)$ with $v_i \in V(G)$ that satisfies $v_i \sim v_{i+1}$ for all $1 \leqslant i < n$, where $\sim$ denotes adjacency between vertices.

Definition 3.1.1 (Excess)**.** Given a multigraph $H = (V, E)$, its *excess* is $\mathrm{exc}(H) = |E| - |V|$.

We can think of excess as the minimum number of edges we can remove from our graph to obtain a tree.

Definition 3.1.2 (A/uni/bi-cyclic)**.** A connected multigraph H with $\mathrm{exc}(H) = -1, 0, 1$ (respectively) is said to be *acyclic*, *unicyclic*, *bicyclic* (respectively). In either of the first two cases, we call H *bicycle-free* (or *at most unicyclic*).

A non-standard definition that will be really important throughout this thesis is the definition of *bicycle-freeness*.

Definition 3.1.3 (Bicycle-free at radius r)**.** We say a multigraph is *bicycle-free at radius* r if the distance-r neighborhood of every vertex is bicycle-free. Another way to say this is that a breadth-first search of depth r, started at any vertex, encounters at most one "back-edge".

The adjacency matrix A_G of G is the matrix with rows and columns indexed by the vertices of G, where for $u, v \in V(G)$, $(A_G)_{uv}$ is equal to the number of occurrences of the edge $\{u, v\}$ in $E(G)$. Given a vector $f : V(G) \to \mathbb{R}$, we have that $(A_G f)_v = \sum_{v \sim u} A_{vu} f_u$. A_G is a real symmetric matrix and thus the spectral theorem tells us that there are n real *eigenvalues* with corresponding orthogonal *eigenvectors*. We order the eigenvalues and denote them by $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_n$ and we write $\mathrm{Spec}(A_G)$ to describe the set of all eigenvalues.

The largest eigenvalue λ_1 is known as the *trivial eigenvalue*. Additionally, we define the *spectral radius* of A_G as $\rho(A_G) = \max_i |\lambda_i|$. The *eigenvalue decomposition* of A_G is a factorization into $U \Lambda U^T$, where U is an orthogonal matrix (meaning $U^T = U^{-1}$) formed from

the eigenvectors of A_G as columns and Λ is a diagonal matrix containing its eigenvalues, i.e. $\Lambda_{ii} = \lambda_i$.

If G is a d-regular graph, then the adjacency matrix A_G has extra structure. Its largest eigenvalue λ_1 is always equal to d, since the vector $\phi_1 = \mathbf{1}/\sqrt{n} = (1/\sqrt{n}, \ldots, 1/\sqrt{n})$ satisfies $A_G\phi_1 = d\phi_1$. If the graph G is disconnected then $\lambda_2 = d$, which means $\lambda(G) = d$. Indeed, consider two connected components C_1 and C_2 and let v_1 be the vector with ones on the elements corresponding to vertices in C_1 and v_2 be the vector with ones on the elements corresponding to vertices in C_2. These two vectors are orthogonal but both satisfy $A_G v_i = dv_i$. It is also known that the smallest eigenvalue λ_n satisfies $\lambda_1 \geqslant -\lambda_n$ (this follows from the Perron-Frobenius theorem [Spi19]).

Let's now briefly consider generic symmetric matrices in $\mathbb{R}^{n\times n}$ with eigenvalues $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_n$. A well known identity on the trace and determinant of such matrices M says that $\mathrm{tr}(M) = \sum_i \lambda_i$ and $\det(M) = \prod_i \lambda_i$. It's also easy to show that the set of eigenvalues of M^k, for any non-negative integer k, is $\{\lambda_1^k, \ldots, \lambda_n^k\}$. There is a variational interpretation of eigenvalues that will be useful to us, which is given by the Courant-Fischer theorem that says the following:

Theorem 3.1.4 (Courant-Fischer)**.** *Let M be a symmetric matrix with eigenvalues $\lambda_1 \geqslant \ldots \geqslant \lambda_n$. Then,*

$$\lambda_k = \max_{\substack{S\subseteq\mathbb{R}^n \\ \dim(S)=k}} \min_{\substack{x\in S \\ x\neq 0}} \frac{x^T M x}{x^T x}.$$

In particular, this implies that λ_1 is the maximum of $x^T M x$ for all vectors $x \in \mathbb{R}^n$ with $\|x\| = 1$. A matrix is *positive semi-definite* (PSD) if all its eigenvalues are non-negative, which by the above is the same as having $x^T M x \geqslant 0$ for all $x \in \mathbb{R}^n$. Given this definition, we can define the *Loewner ordering* of symmetric matrices as a partial order $\succeq$ such that for symmetric matrices $A, B \in \mathbb{R}^n$ we have $A \succeq B$ if $A - B$ is PSD.

For a thorough reference on the fundamentals of spectral graph theory, including proofs of the theorems and properties sated above, we recommend the book of Spielman [Spi19].

3.2 The trace method, non-backtracking walks and the Ihara–Bass formula

We saw in Chapter 1 that our main focus will be to study the spectral expansion $\lambda(G)$ of graphs G. Namely, we will want to either upper bound the spectral expansion of some given (random) graphs or construct graphs that have some upper bound on this quantity.

A common tool to bound the largest eigenvalue of a symmetric matrix is the Füredi-Komlós Trace Method [FK81]. Let H be a $n \times n$ symmetric real matrix and denote its eigenvalues by $\lambda_1 \geqslant \ldots \geqslant \lambda_n$. Then, for any non-negative integer k, we know that $\sum_{i=1}^n \lambda_i^k = \mathrm{tr}(H^k)$. If k is an even integer, we have that $\lambda_1^k \leqslant \mathrm{tr}(H^k)$. Suppose we can bound each element of the diagonal of H^k by $a(k)$. The idea of the Trace Method is to take $k \gg \log n$ to conclude that $\lambda_1 \leqslant n^{1/k} a(k)^{1/k} = (1 + o_n(1))a(k)^{1/k}$.

More generally, the following inequality is applied when using this technique:

$$\operatorname{tr}(A^{2k}) = \operatorname{tr}((A^T)^k A^k) = \|A^k\|_F^2 = \sum_i |\lambda_i^k|^2 \geqslant \rho(A)^{2k},$$

where $\|A\|_F = \sqrt{\sum_{i,j} A_{ij}^2}$ is the Frobenius norm of A.

The reason why this inequality is so powerful is because we can now study the spectrum of A (namely its spectral radius $\rho(A)$) through $\operatorname{tr}(A^{2k})$. Notice that A^{2k} is the matrix of length $2k$ walks in G, that is, $(A^{2k})_{ij}$ is exactly the number of walks of length $2k$ from i to j in G. Hence $\operatorname{tr}(A^{2k}) = \sum_i (A^{2k})_{ii}$ is the total number of closed walks in G (walks that start and end at the same vertex). By using this method we can analyze a spectral property of G (i.e. $\rho(A)$) through purely combinatorial properties of G (i.e. the number of closed walks).

So we have seen how using the Trace Method to analyze eigenvalues of graphs involves counting closed walks. However, it is common to instead count *non-backtracking* walks, since these are much easier to count. These are walks that never backtrack, that is, we never take the same edge twice in a row (but we might repeat it later in the walk). As it turns out, we can relate the number of closed walks to the number of non-backtracking walks. Hence, we define the *non-backtracking matrix*.

Definition 3.2.1 (Non-backtracking matrix [Has89])**.** Let $G = (V, E)$ be a multigraph with adjacency matrix A. Let $\vec{E}$ denote the (multi)set of all directed edges formed by replacing each undirected edge in E with two opposing directed edges. Then G's *non-backtracking* matrix B has rows and columns indexed by $\vec{E}$, with

$$B_{(u_1,v_1),(u_2,v_2)} = \begin{cases} A_{u_2,v_2} & \text{if } v_1 = u_2 \text{ and } v_2 \neq u_1, \\ 0 & \text{otherwise.} \end{cases}$$

(Note that this matrix is not symmetric in general.) In case G is an edge-signed graph, the entry 1 above should be replaced by A_{u_2,v_2}, the sign of G on edge $\{u_2, v_2\}$.

Ultimately, we want a map between eigenvalues of this matrix B and the adjacency matrix A. In a number-theoretic context, Ihara [Iha66] implicitly showed how to do so when the graph G is regular. Serre [Ser77] and several others suggested the translation to graph theory, and Bass [Bas92] (following [Has89]) explicitly established:

Theorem 3.2.2. *(Ihara–Bass formula.) Let G be a d-regular (multi)graph and write $q = d - 1$. Then*

$$\det(\mathbb{1} - zB) = (1 - z^2)^{\operatorname{exc}(G)-1} \det((1 + qz^2)\mathbb{1} - zA),$$

where $\mathbb{1}$ denotes the identity matrix (of appropriate dimension).

This theorem has been given many proofs, and it can be generalized to irregular graphs, edge-weighted graphs, and infinite graphs, see [WF09, AFH15]. We will use the following result, which is immediate from the edge-weighted generalization [WF09] when all weights are ± 1:

Theorem 3.2.3. *([WF09].) The Ihara–Bass formula holds as stated above for edge-signed graphs.*

The utility of Ihara–Bass is that it gives a direct correspondence between the spectra of A and B. To see this, consider the zeroes of the polynomials (in z) on the left- and right-hand sides. We have that z is a zero of the left-hand side precisely if z^{-1} is an eigenvalue of B. On the other

hand, z is a zero of the right-hand side precisely if $z^{-1} = \pm 1$ or if z^{-1} is such that $z^{-1} + q/z^{-1}$ is an eigenvalue of A. Thus if we want to deduce, say, the eigenvalues of B from the eigenvalues of A, we have the following:

Proposition 3.2.4. *(Consequence of Ihara–Bass.) Let $G = (V, E)$ be a $(q+1)$-regular edge-signed graph with adjacency matrix A and non-backtracking matrix B. Let $\lambda \neq 0, \pm 1$ be a number such that $\lambda + q/\lambda$ is an eigenvalue of A. Then λ is an eigenvalue of B.*

In fact, Proposition 3.2.4 is the only consequence of Ihara–Bass we will need, and for the convenience of the reader we give a self-contained proof (inspired by [AFH15]):

Proof. Let $f : V \to \mathbb{C}$ be an eigenvector for A with eigenvalue $\lambda + q/\lambda$. Define $g : \vec{E} \to \mathbb{C}$ by $g_{vw} = A_{vw} f_v - \lambda f_w$. We claim that $Bg = \lambda g$. It then follows that λ is an eigenvalue of B, given that $g \not\equiv 0$ (a consequence of $f \not\equiv 0$: choose $\{v, w\} \in E$ with f_v, f_w not both 0, and then $g_{vw} = 0 = g_{wv}$ is impossible because $\lambda \neq \pm 1$). To verify the claim, for any $uv \in \vec{E}$ we have

$$(Bg)_{uv} = \sum_{\substack{w \sim v \\ w \neq u}} A_{vw} g_{vw} = \sum_{w \sim v} A_{vw}(A_{vw} f_v - \lambda f_w) - A_{vu}(A_{vu} f_v - \lambda f_u) = -\lambda \sum_{w \sim v} A_{vw} f_w + q f_v + \lambda A_{vu} f_u.$$

But $\sum_{w \sim v} A_{vw} f_w = (Af)_v = (\lambda + q/\lambda) f_v$. Thus $(Bg)_{uv} = -\lambda^2 f_v + \lambda A_{vu} f_u = \lambda g_{uv}$, as needed. □

When G is unsigned, A has a "trivial" eigenvalue of $d = q + 1$, corresponding to $\lambda = q$; this yields the "trivial" eigenvalue of $q = d - 1$ for B. For general edge-signed G, if $\lambda = \pm\sqrt{q} = \pm\sqrt{d-1}$ in Proposition 3.2.4, then $\lambda + q/\lambda = \pm 2\sqrt{q} = \pm 2\sqrt{d-1}$. Thus the Ramanujan eigenvalue bound of $2\sqrt{d-1}$ for A is equivalent to the bound $\sqrt{d-1}$ for B. As for the "$+\varepsilon$", a simple calculation (appearing in [Bor19], Section 2.2) shows:

Corollary 3.2.5. *Let $G = (V, E)$ be a d-regular graph ($d \geqslant 3$) with adjacency matrix A and non-backtracking matrix B. If A has an eigenvalue of magnitude $2\sqrt{d-1} + \varepsilon$ (for $\varepsilon \geqslant 0$) then B has an eigenvalue of magnitude $\sqrt{d-1} + \sqrt{\varepsilon}\sqrt{\sqrt{q} + \varepsilon/4} + \varepsilon/2$ (which is $\sqrt{d-1} + \Theta(d^{1/4}\sqrt{\varepsilon})$ for fixed d and $\varepsilon \to 0$).*

3.3 Random models of regular graphs

We will devote a lot of our time to analyze properties of random regular graphs. As such, we will need to look at some of the classic models. We start by describing one standard way to generate random d-regular graphs: the *random lift model*, see [BC78, Bol80, Bol01].

Definition 3.3.1 (Lift model)**.** Fix a *base* graph $\underline{G} = (\underline{V}, \underline{E})$ on $\underline{n}$ vertices. Then for $n \in \mathbb{N}^+$, an *n-lift of $\underline{G}$* is graph G defined by a collection of permutations $\pi_{uv} \in S_n$, one for each edge $(u, v) \in \vec{\underline{E}}$, under the constraint that $\pi_{uv} = \pi_{vu}^{-1}$. The vertex set of G is $\underline{V} \times [n]$, and the edges of G are given by all pairs $(u, i), (v, j)$ satisfying $(u, v) \in \underline{E}$ and $\pi_{uv}(i) = j$. When the permutations π_{uv} are independent and uniformly random, we call the associated graph $\boldsymbol{G}$ a *(uniformly) random n-lift of $\underline{G}$*. Observe that if $\underline{G}$ is a d-regular graph, then $\boldsymbol{G}$ is always a d-regular (simple) graph on $\underline{n}n$ vertices.

A simple observation is that if $\underline{G}$ is a d-regular graph, then any graph lift of $\underline{G}$ is a d-regular graph on $|V(\underline{G})|n$ vertices. An important case of the lift model is the one of *2-lifts*. An equivalent way of defining a 2-lift is by considering an edge-signing $w : \underline{E} \to \{\pm 1\}$ of $\underline{G}$. This edge-signing uniquely defines a 2-lift of $\underline{G}$, which we can describe in the following way:

$$V = \underline{V} \times \{\pm 1\}, \qquad E = \Big\{\{(u,\sigma),(v,\sigma \cdot w(u,v))\} : (u,v) \in \underline{E}, \sigma \in \{\pm 1\}\Big\}.$$

The following was first observed by Bilu and Linial [BL06]:

Lemma 3.3.2. *Let $\underline{G}$ be a d-regular graph, $w : \underline{E} \to \{\pm 1\}$ an edge-signing and $\widetilde{\underline{G}}$ the signed version of $\underline{G}$ (meaning the graph such that its adjacency matrix has nonzero entries $w(u,v)$ when $\{u,v\} \in \underline{G}$). Then the corresponding 2-lift G satisfies:*

$$\mathrm{Spec}(A_G) = \mathrm{Spec}(A_{\underline{G}}) \cup \mathrm{Spec}(A_{\widetilde{\underline{G}}}).$$

We can now define our second regular graph model.

Definition 3.3.3 (Configuration model)**.** Given integers $n > d > 0$ with nd even, the *configuration model* produces a random n-vertex, d-regular undirected multigraph (with loops) $\boldsymbol{G}$. This multigraph is induced by a uniformly random matching $\boldsymbol{M}$ on the set of "half-edges", $[n] \times [d] \cong [nd]$ (where $(v,i) \in [n] \times [d]$ is thought of as half of the ith edge emanating from vertex v). We identify $\boldsymbol{M}$ with a symmetric matrix in $\{0,1\}^{nd \times nd}$ having 1's precisely in the entries corresponding to matched pairs $\{(v,i),(v',i')\}$. We may think of $\boldsymbol{M}$ being generated as follows: First a uniformly random permutation $\boldsymbol{\pi} \in S_{nd}$ is chosen; then we set $\boldsymbol{M}_{\boldsymbol{\pi}(j),\boldsymbol{\pi}(j+1)} = \boldsymbol{M}_{\boldsymbol{\pi}(j+1),\boldsymbol{\pi}(j)} = 1$ for each odd $j \in [nd]$.

Given $\boldsymbol{M}$, the multigraph $\boldsymbol{G}$ is formed by "attaching" the matched half-edges. More formally, the (v,v')-entry of $\boldsymbol{G}$'s adjacency matrix $\boldsymbol{A}$ is the sum, over all $i,i' \in [d]$, of $\boldsymbol{M}_{(v,i),(v',i')}$. Hence

$$\boldsymbol{A}_{v,v'} = \sum_{i,i'=1}^{d} \sum_{\substack{\text{odd} \\ j \in [nd]}} (1[\boldsymbol{\pi}(j) = (v,i)] \cdot 1[\boldsymbol{\pi}(j+1) = (v',i')] + 1[\boldsymbol{\pi}(j) = (v',i')] \cdot 1[\boldsymbol{\pi}(j+1) = (v,i)]).$$

Note that $\boldsymbol{A}_{v,v}$ will always be even; a self-loop is considered to contribute degree 2.

It is well known that a graph $\boldsymbol{G}$ drawn from the configuration model is simple [Wor99] — i.e., has no cycles of length 1 or 2 — with probability $\Omega_d(1)$, this continues to hold for *pseudorandom* d-regular graphs (to be defined later.) We also record the well known fact that for $\boldsymbol{G}$ drawn from the configuration model, when $\boldsymbol{G}$ is conditioned on being simple, its conditional distribution is uniformly random among all d-regular graphs.

It is easy to see that any n-vertex, d-regular graph that is bicycle-free at radius r must have $r \lesssim \log_{d-1} n$. On the other hand, it can be shown (see, for example, [Bor19]) that a random d-regular graph achieves this bound up to a constant factor.

3.4 Standard derandomization tools

All of the explicit constructions we will present follow a simple recipe: first show some property about random graphs (drawn from the appropriate distribution); then derandomize this property;

apply a deterministic formula to the result to obtain the desired explicit construction. As such, we will need to use several tools from the pseudorandomness literature, which we will describe here.

Definition 3.4.1 ((δ, k)-wise uniform bits)**.** Let $\delta \in [0, 1]$ and $k \in \mathbb{N}^+$. A sequence of Boolean random variables $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) \in \{\pm 1\}^n$ is said to be (δ, k)*-wise uniform*[1] if, for every $S \subseteq [n]$ with $0 < |S| \leqslant k$, it holds that $|\mathbf{E}[\prod_{i \in S} \boldsymbol{y}_i]| \leqslant \delta$. When $\delta = 0$, we simply say that the sequence is *(truly)* k*-wise uniform*; indeed, in this case the bits are individually uniformly distributed and are k-wise independent.

A classic result of Naor and Naor [NN93] shows that (δ, k)-wise uniform bits can be constructed efficiently and deterministically from a truly random seed of length $O(\log k + \log\log n + \log(1/\delta))$. Indeed, these bits can be generated "strongly explicitly" (using [Sho90]; cf. [AGHP92]):

Theorem 3.4.2. *([NN93].) There is a deterministic algorithm that, given δ, k, and N, runs in time $\mathrm{poly}(N/\delta)$ and outputs a multiset $Y \subseteq \{\pm 1\}^N$ of cardinality $S = \mathrm{poly}(k \log(N)/\delta)$ (a power of 2) such that, for $\boldsymbol{y} \sim Y$ chosen uniformly at random, the sequence $\boldsymbol{y}$ is (δ, k)-wise uniform. Indeed, if the algorithm is additionally given $1 \leqslant s \leqslant S$ and $1 \leqslant i \leqslant N$ (written in binary), it can output the ith bit of the sth string in Y in deterministic time $\mathrm{polylog}(N/\delta)$.*

We will make use of the fact that the parameters in this theorem have excellent dependence on N and k. We now discuss the analogous concept for random permutations, where it is not known if the parameter dependence can be as strong.

Definition 3.4.3 ((δ, k)-wise uniform permutations)**.** Let $\delta \in [0, 1]$ and $k \in \mathbb{N}^+$. Let $[n]_k$ denote the set of all sequences of k distinct indices from $[n]$. A random permutation $\boldsymbol{\pi} \in S_n$ is said to be (δ, k)*-wise uniform* if, for every sequence $(i_1, \ldots, i_k) \in [n]_k$, the distribution of $(\boldsymbol{\pi}(i_1), \ldots, \boldsymbol{\pi}(i_k))$ is δ-close in total variation distance from the uniform distribution on $[n]_k$. When $\delta = 0$, we simply say that the permutation is *(truly)* k*-wise uniform*.

Kassabov [Kas07] and Kaplan–Naor–Reingold [KNR09] independently obtained a deterministic construction of (δ, k)-wise uniform permutations with seed length $O(k \log n + \log(1/\delta))$. Again, the construction is even "strongly explicit":

Theorem 3.4.4. *([KNR09, Kas07].) There is a deterministic algorithm that, given δ, k, and n, runs in time $\mathrm{poly}(n^k/\delta)$ and outputs a multiset $\Pi \subseteq S_n$ (closed under inverses) of cardinality $S = \mathrm{poly}(n^k/\delta)$ (a power of 2) such that, for $\boldsymbol{\pi} \sim \Pi$ chosen uniformly at random, $\boldsymbol{\pi}$ is a (δ, k)-wise uniform permutation. Indeed, if the algorithm is additionally given $1 \leqslant s \leqslant S$ and $1 \leqslant i \leqslant n$ (written in binary), it can output $\pi_s(i)$ and $\pi_s^{-1}(i)$ (where π_s is the sth permutation in Π) in deterministic time $\mathrm{poly}(k \log(n/\delta))$.*

We will also use a convenient theorem of Alon and Lovett [AL13]:

Theorem 3.4.5. *([AL13].) Let $\boldsymbol{\pi} \in S_n$ be a (δ, k)-wise uniform permutation. Then one can define a (truly) k-wise uniform permutation $\boldsymbol{\pi}' \in S_n$ such that the total variation distance between $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ is $O(\delta n^{4k})$.*

Combining the previous two results yields the following:

Corollary 3.4.6. *([KNR09, Kas07, AL13]) There is a deterministic algorithm that, given k and n, runs in time $\mathrm{poly}(n^k)$ and outputs a multiset $\Pi \subseteq S_n$ (closed under inverses) such that, when*

[1] Frequently called (δ, k)-wise *independent* in the literature.

$\boldsymbol{\pi} \sim \Pi$ is chosen uniformly at random, $\boldsymbol{\pi}$ is n^{-100k}-close in total variation distance to a (truly) k-wise uniform permutation. (And the final "indeed" statement from Theorem 3.4.4 also holds.)

3.5 A primer on coding theory

Some of our applications of expanders are to the field of *coding theory*. We provide a (really) short introduction to the main terms of this field. For a more comprehensive view of this field we recommend the book [GRS12].

Suppose there are two parties that want to communicate information through some channel. For example, the two parties can be two friends and the channel the internet. However, suppose the channel is noisy, that is often the information being transmitted gets (partially) corrupted. We need a way to represent data such that we can recover the original message even if part of it gets corrupted. This is exactly what *error correcting codes* do.

Formally, an error correcting code $\mathcal{C}$ of *code length* n over an *alphabet* Σ is a subset of Σ^n. Elements of $\mathcal{C}$ are known as *codewords*. We usually represent the cardinality of the code $|\Sigma|$ by q. The *dimension* of the code is given by $\log_q |\mathcal{C}|$, which we usually denote by k or $\dim \mathcal{C}$. An alternate way of defining an error correcting code is as an injective map $\Sigma^k \to \Sigma^n$.

A *linear code* is a code for which any linear combination of codewords is also a codeword. Consider a finite field $\mathbb{F}_q$ and an n-dimensional vector space $\mathbb{F}_q^n$ over $\mathbb{F}_q$. Formally, a linear $[n,k]_q$ code is a k-dimensional subspace of $\mathbb{F}_q^n$. We can associate to $\mathcal{C}$ a full-rank $n \times k$ matrix G, called the *generator matrix*, such that $\mathcal{C}$ is the row space of G. Using the map definition of codes, this is equivalent to saying that the injective map $\mathbb{F}_q^k \to \mathbb{F}_q^n$ that defines the code is given by $x \mapsto Gx$. Additionally, we can associate a $(n-k) \times n$ matrix H, called the *parity check matrix*, such that $\mathcal{C}$ is the kernel of H, which means that for $x \in \mathcal{C}$ we have $Hx = 0$. It is not hard to see that $GH^T = 0$.

The *dual code* of a code $\mathcal{C}$, denoted by $\mathcal{C}^\perp$, is a $[n, n-k]_q$ linear code defined as $\mathcal{C}^\perp = \{x \in \mathbb{F}_q^n | \forall y \in \mathcal{C}, \langle x, y \rangle = 0\}$. It is easy to see that the generator matrix for $\mathcal{C}$ is H and the parity check matrix is G.

The *Hamming distance* between two vectors $x, y \in \mathbb{F}_q^n$, denoted by $d(x,y)$, is the number of positions at which the corresponding symbols are different. The *distance of a code* $\mathcal{C}$ is the minimum Hamming distance between distinct codewords, formally $d(\mathcal{C}) = \min\{d(x,y) | x \neq y; x, y \in \mathcal{C}\}$.

A *low density parity check (LDPC) code* is a linear code whose parity check matrix has row and column weights bounded by a constant w. They were first introduced by Gallager [Gal62] in the '60s and are one of the most popular classes of classical error-correcting codes, both in theory and in practice. This popularity comes from the fact that there are many known constructions of classical LDPC codes that achieve linear rate and distance that can also be decoded in linear time [RU08].

Chapter 4

2-Lifts and Explicit Near-Ramanujan Graphs

We begin this thesis by studying the oldest and most fundamental problem in the area of expanders: how to explicitly construct (near) optimal spectral expander graphs. The tools we introduce in this chapter are used repeatedly in the remaining chapters.

We describe the results of [MOP20a], which was joint work with Sidhanth Mohanty and Ryan O'Donnell. We showed how to construct near-Ramanujan graphs in a probabillistically strongly explicit way.

Our proof has two parts. First, we weakly derandomize Bordenave's proof of Theorem 1.1.16 to produce a small d-regular near Ramanujan graph, using standard derandomization tools. Additionally, we show how to make this small graph be bicycle-free at a large enough radius. Then, we show that a random 2-lift of a graph that is bicycle-free at such a radius is also near-Ramanujan, which we also derandomize. Finally, by iterating this procedure we are able to obtain a near-Ramanujan graph of the desired size.

4.1 Overview of main results

Our main result of this chapter gives $\mathrm{poly}(n)$-time deterministically computable n-vertex d-regular graphs G with $\lambda(G) \leqslant 2\sqrt{d-1} + \varepsilon$, for any $d \geqslant 3$ and $\varepsilon > 0$. To be more precise, the running time of our algorithm is $n^{f(d,\varepsilon)}$ where $f(d,\varepsilon) = O(d^{1/4}\log(d)/\sqrt{\varepsilon})$. Although our graphs are not strongly explicit, they are at least probabilistically strongly explicit. Recall that this means we show there *exist* near-Ramanujan graphs whose adjacency lists are computable in $\mathrm{polylog}(n)$ time, and furthermore there is a $\mathrm{polylog}(n)$-time randomized algorithm for finding them with high probability. More precisely, the following statement holds:

Theorem 4.1.1. *There is a deterministic polynomial-time algorithm with the following properties:*

- *It takes as input N, $d \geqslant 3$, and $\varepsilon > 0$ written as binary strings.*
- *It also takes as input a "seed" $s \in \{0,1\}^{O(\log N)}$ (the $O(\cdot)$ hides a factor of $O(d^{1/4}\log(d)/\sqrt{\varepsilon})$).*
- *It outputs a Boolean circuit C that implements the "adjacency list" of a d-regular graph G on $N' \sim N$ vertices in $\mathrm{polylog}(N)$ time. (This means that on input $u \in [N']$ and $i \in [d]$,*

both expressed in binary, $C(u,i)$ outputs the $v \in [N']$ that is the ith neighbor of u in G.)

- *With high probability over the choice of seed s, the resulting graph G satisfies the bound $\lambda(G) \leqslant 2\sqrt{d-1}+\varepsilon$.*

Given Theorem 4.1.1, we can obtain a deterministic polynomial-time ("weakly explicit") construction by taking $N = n$, enumerating all possible $\mathrm{poly}(n)$ seeds s, explicitly constructing each resulting G, and then selecting any of the many ones with $\lambda(G) \leqslant 2\sqrt{d-1}+\varepsilon$. This selection uses the following fact:

Fact 4.1.2. *For any rational approximation ρ of $2\sqrt{d-1}+\varepsilon$, one can decide in $\mathrm{poly}(n)$ time whether $\lambda(G) \leqslant \rho$* [1].

We summarize this fact about the weakly explicit construction in the following corollary:

Corollary 4.1.3. *For any given constants $N, d \geqslant 3, \varepsilon > 0$, there is a deterministic polynomial ($n^{f(d,\varepsilon)}$ where $f(d,\varepsilon) = O(d^{1/4}\log(d)/\sqrt{\varepsilon})$) time algorithm that produces a d-regular graph G with N' vertices such that:*

- $N' \sim N$,
- $\lambda(G) \leqslant 2\sqrt{d-1}+\varepsilon$.

The key technical result that we prove in service of this is the following:

Theorem 4.1.4. *Let G be an arbitrary d-regular n-vertex graph. Assume that G is r-bicycle free, where $r \gg (\log\log n)^2$. Then a random edge-signing of G has all its eigenvalues bounded in magnitude by $2\sqrt{d-1}+o_n(1)$, with high probability.*

4.1.1 On Bordenave's theorem with random edge-signs

Since our result may be viewed as a derandomization of the Friedman/Bordenave theorem (Theorem 1.1.16), let us take some time to describe this result. Friedman's original proof is notably quite involved (100 pages). Bordenave's proof is certainly simpler (more like 30 pages), although it is by no means easy. However, Bordenave's proof can become still simpler if one is willing consider a variant: when $\boldsymbol{G}$ is not just a random d-regular graph, but rather a *randomly edge-signed* random d-regular graph.

Let us say a few words about why this makes things simpler. First, it turns out that in this case one need not worry about the "trivial eigenvalue" of d; it no longer exists, and the statement to be proven is simply that $\rho(\boldsymbol{G}) \leqslant 2\sqrt{d-1}+\varepsilon$ with high probability, where $\rho(\boldsymbol{G})$ is the spectral radius (largest eigenvalue-magnitude) of the (signed) adjacency matrix of $\boldsymbol{G}$. Second, with random edge-signs, each entry of $\boldsymbol{G}$'s adjacency matrix becomes a symmetric random variable, and it is always more pleasant in probability theory when one's random variables naturally have mean zero.

In fact, there are scenarios in which one might actually *want* to consider random edge-signed d-regular graphs. For example when studying the Max-Cut problem, the setting of sparse random graphs is a very natural and challenging one, and many algorithms/complexity results depend on eigenvalue bounds for such graphs. Having random edge-signs simply means studying the equally natural 2XOR (aka 2Lin) problem, one that has a long history in theoretical computer science as well [Hås84].

[1] This is a known fact since it easily reduces to checking in polynomial time whether a given rational matrix is PSD [GLS84, GLS93]

Undoubtedly experts would know that including random edge-signs should make Bordenave's proof simpler, but it doesn't appear to have been directly explored until the recent work of Deshpande et al. [DMO+19a]. That paper proved the analogue of Friedman/Bordenave for random edge-signings of random (c, d)-biregular graphs. The case when $c = d$ is essentially the same as the d-regular random graph case, but the nature of the proof simplification is perhaps obscured, particularly because [DMO+19a] directly cited several lemmas from Bordenave [Bor19].

In fact, a side motivation we had was to carefully set out a self-contained proof — as simple as possible — of "Alon's Conjecture" for randomly edge-signed graphs. A reader not interested in derandomization may nevertheless find our proof of the below theorem of interest, particularly since it contains a substantial portion of Bordenave's proof of Friedman's theorem.

Theorem 4.1.5. *Let $d \geqslant 3$ and $\varepsilon > 0$. If $\boldsymbol{G}$ is a random edge-signed d-regular n-vertex graph, then*

$$\mathbf{Pr}\Big[\rho(\boldsymbol{G}) \leqslant 2\sqrt{d-1} + \varepsilon\Big] \geqslant 1 - o_n(1).$$

In the course of proving this theorem, we are able to observe that in fact Theorem 4.1.4 holds. That is, Theorem 4.1.5 does not thoroughly rely on having a random edge-signing of a *random* d-regular graph. Instead, it works for a random edge-signing of *any* d-regular graph that has one particular property: namely, every vertex-neighborhood of radius $O((\log\log n)^2)$ should have at most one cycle. This property — called tangle-freeness by Bordenave (simplifying Friedman's notion of "tangles") — is a property that random d-regular graphs have with high probability, even for neighborhoods of the much larger radius $\Theta(\log_{d-1} n)$.

With Theorem 4.1.4 in hand, we are in a position rather like that of Bilu–Linial, who similarly showed [BL06, Cor. 3.1] that a random edge-signing of any sufficiently good small-set expander has spectral radius at most $\sqrt{d} \cdot O(\log^{1.5} d)$ (with high probability). As in Bilu–Linial, it is also fairly straightforward to see that Theorem 4.1.4 can be derandomized effectively using almost-k-wise independent binary random variables.

We next describe how this derandomized result on edge-signings leads to our main Theorem 4.1.1.

4.1.2 Explicit near-Ramanujan graphs via repeated 2-lifts

Let $G = (V, E)$ be an n-vertex d-regular graph, and let $\widetilde{G}$ be the edge-signed version of it associated to edge-signing $w : E \to \{\pm 1\}$. As observed by Bilu and Linial [BL06], this edge-signing is in a sense equivalent to the "2-lift" $G_2 = (V_2, E_2)$ of G defined by

$$V_2 = V \times \{\pm 1\}, \qquad E_2 = \Big\{\{(u, \sigma), (v, \sigma \cdot w(u, v))\} : (u, v) \in E\Big\}.$$

This G_2 is a $2n$-vertex d-regular graph, and the equivalence is that G_2's eigenvalues are precisely the multiset-union of G's eigenvalues and $\widetilde{G}$'s eigenvalues. (The latter refers to the eigenvalues of $\widetilde{G}$'s signed adjacency matrix, whose nonzero entries are $w(u, v)$ for each $\{u, v\} \in E$.) In particular, if all the eigenvalues of G and $\widetilde{G}$ have magnitude at most $2\sqrt{d-1} + \varepsilon$ (excluding G's trivial eigenvalue of d), then the same is true of G_2 (excluding *its* trivial eigenvalue). Thus Theorem 4.1.4 can provide us with a (derandomizable) way of doubling the number of vertices in an ε-near-Ramanujan graph. It is not hard to see (Proposition 4.2.1) that if G is r-bicycle-free

then G_2 will also be r-bicycle-free. Thus we may repeatedly double the number of vertices in an ε-near-Ramanujan graph, so long as the parameter r remains $\omega((\log\log|V|)^2)$, where $|V|$ is the "current" number of vertices. (Unfortunately, we do not see an obvious way to get the parameter r to increase as we perform 2-lifts.) This is roughly the same strategy employed in [BL06].

As a consequence, to obtain a final d-regular ε-near-Ramanujan graph with $\Theta(N)$ vertices, all we need to get started is some d-regular ε-near-Ramanujan graph H on a smaller number of vertices, n, which is $O((\log\log N)^2)$-bicycle-free. Thanks to Friedman/Bordenave, we know that a *random* d-regular n-vertex graph is (with high probability) near-Ramanujan, and it's not hard to show it's $\Theta(\log n)$-bicycle-free. Thus we could get started with H being a random d-regular graph on, say, $n = 2^{\sqrt{\log N}}$ vertices, or even something smaller like $n = \mathrm{quasipoly}(\log N)$.

Of course, to get a construction which is overall explicit, we need to derandomize the Friedman/Bordenave analysis for this base graph H. The advantage is we now have $\mathrm{poly}(N)$ time to spend on constructing a graph with $n \ll N$ vertices. A trivial exponential-time derandomization won't work, but nor do we need a polynomial-time derandomization; a quasipolynomial-time derandomization is more than sufficient. And as we will see in Section 4.4, it is possible to derandomize Bordenave's proof in deterministic $n^{O(\log n)}$ time using $O(\log n)$-wise uniform permutations. The proof of this is not completely straightforward because Bordenave's proof uses a twist on the Füredi-Komlós Trace Method [FK81] (since the plain Trace Method provably fails).

4.2 On bicycle-freeness

It is well known that a d-regular random graph in the random lift model is likely to have at most one cycle in any neighborhood of radius $c\log_{d-1} n$, for a certain universal $c > 0$. To codify this, let us first recall the definition of *bicycle-freeness*:

Definition 3.1.3 (Bicycle-free at radius r)**.** We say a multigraph is *bicycle-free at radius* r if the distance-r neighborhood of every vertex is bicycle-free. Another way to say this is that a breadth-first search of depth r, started at any vertex, encounters at most one "back-edge".

Proposition 4.2.1. *If G is bicycle-free at radius r, and G_2 is a 2-lift of G, then G_2 is bicycle-free at radius r.*

Proof. Let (v,i) be any vertex in G_2. Let H be the distance-r neighborhood of v in G and let H_2 be the subgraph of G_2 induced by $V(H)\times[2]$. Observe that the distance-r neighborhood of (v,i) is contained in H_2, and that $\mathrm{exc}(H_2) \leqslant 0$ since $\mathrm{exc}(H) \leqslant 0$. If H_2 is disconnected it is isomorphic to a disjoint union of two copies of H and thus the distance-r neighborhood of (v,i) is then isomorphic to H. Otherwise, if H_2 is connected, $\mathrm{exc}(H_2) \leqslant 0$ implies that it has at most one cycle. □

It is easy to see that any n-vertex, d-regular graph that is bicycle-free at radius r must have $r \lesssim \log_{d-1} n$. On the other hand, as mentioned earlier, a random d-regular graph achieves this bound up to a constant factor, and we will derandomize the proof of this fact, within the $O(\log n)$-wise uniform lift model, in Section 4.4.1.

In a graph that is bicycle-free at radius r, by definition we have $\mathrm{exc}(H) \leqslant 0$ for all subgraphs H contained in a single distance-r neighborhood. In fact, this property is enough to

guarantee that $\mathrm{exc}(H)$ is small for *any* subgraph H with at most $\exp(r)$ vertices, regardless of whether it's contained in a single distance-r neighborhood:

Theorem 4.2.2. *Let H be a v-vertex graph that is bicycle-free at radius r. Assume $r \geqslant 10 \ln v$. Then $\mathrm{exc}(H) \leqslant \frac{\ln(ev)}{r} v$.*

The rest of this subsection is devoted to the proof of the above theorem of elementary graph theory.

Definition 4.2.3 ($\mathrm{Cyc}_g(G)$ and girth)**.** Given a graph G, let $\mathrm{Cyc}_g(G)$ denote the collection of all cycles in G of length at most g. Recall that if $\mathrm{Cyc}_g(G)$ is empty then G is said to have *girth* exceeding g.

The following fact is essentially immediate from the definitions:

Fact 4.2.4. *Suppose G is bicycle-free at radius r. Then the cycles in $\mathrm{Cyc}_{2r}(G)$ are vertex-disjoint.*

Indeed, more generally:

Proposition 4.2.5. *Suppose G is bicycle-free at radius r. For each $C \in \mathrm{Cyc}_{2r}(G)$, let C^+ denote the collection of vertices within distance $r - \mathrm{len}(C)/2$ of C. Then the sets $\{C^+ : C \in \mathrm{Cyc}_{2r}(G)\}$ are pairwise disjoint.*

Proof. If $u \in C_1^+ \cap C_2^+$, the distance-r neighborhood of u is enough to include both C_1 and C_2. □

Next, let us now recall the "Moore bound for irregular graphs". Suppose H is a graph with v vertices and $\mathrm{exc}(H) = \varepsilon v$; hence H has average degree $2 + 2\epsilon$. If we build a breadth-first search tree from some vertex, then after depth t we would "expect" to encounter at least $(1 + 2\epsilon)^t$ vertices. If this exceeds v — roughly, if $t \geqslant (\ln v)/(2\epsilon)$ — then the breadth-first search must encounter a cycle. Thus we have a heuristic argument that $\mathrm{girth}(H) \lesssim (\ln v)/\epsilon$; i.e., $\epsilon \lesssim (\ln v)/\mathrm{girth}(H)$. Indeed, Alon–Hoory–Linial have precisely established this kind of result; we quote their theorem in a slightly simplified form:

Theorem 4.2.6. *([AHL02].) Let H be a graph with v vertices, $\mathrm{exc}(H) = \varepsilon v$ (for $\varepsilon \geqslant 0$), and girth g. Then $v \geqslant (1 + 2\varepsilon)^{g/2-3/2}$.*

Corollary 4.2.7. *Let H be a graph with $v \geqslant 3$ vertices and girth $g \geqslant 20 \ln v$. Then $\mathrm{exc}(H) \leqslant ((2 \ln v)/g)v$.*

Proof. Suppose for the sake of contradiction that $\mathrm{exc}(H) > ((2 \ln v)/g)v$. Now we apply Theorem 4.2.6 and take logs to obtain:

$$\begin{aligned}
&\ln v > \frac{g-3}{2} \cdot \ln\left(1 + \frac{4 \ln v}{g}\right) \\
\Rightarrow &2 \ln v > (g-3) \cdot \frac{4(\ln v)/g}{1 + 4(\ln v)/g} \\
\Rightarrow &\frac{1}{2}(g + 4 \ln v) + 3 > g \\
\Rightarrow &20 \ln v > g,
\end{aligned}$$

where the first implication uses the inequality $\ln(1 + x) \geqslant x/(x + 1)$ for $x > -1$. □

We can now prove Theorem 4.2.2, which replaces "girth" with "bicycle-free radius" in the above with only a small loss in parameters.

Proof of Theorem 4.2.2. We will show the theorem assuming H is connected (the only case we'll need). It is an exercise to extend it to the general case by considering H's connected components.

Let $c = |\mathrm{Cyc}_{2r}(H)|$. By deleting at most c edges from H we can obtain a v-vertex graph $\widetilde{H}$ with girth at least (in fact, exceeding) $2r$. Applying Corollary 4.2.7 to $\widetilde{H}$, we conclude that $\mathrm{exc}(H) \leqslant \frac{\ln v}{r} v + c$. Thus it remains to show $c \leqslant v/r$. This is trivial if $c = 0$, and if $c = 1$ then it can only fail if $r > v$ — but then H is unicyclic and hence has excess 0. Assuming then that $c \geqslant 2$, choose paths in H to minimally connect the c cycles of $\mathrm{Cyc}_{2r}(H)$. Now for each $C \in \mathrm{Cyc}_{2r}(H)$, if we "charge" to it the $r - \mathrm{len}(C)/2$ closest path-vertices, then no vertex is charged by multiple cycles, by virtue of Proposition 4.2.5. If we also charge the vertices of C to itself, then for each $C \in \mathrm{Cyc}_{2r}(H)$ we have charged a batch of at least $\mathrm{len}(C) + (r - \mathrm{len}(C)/2) > r$ vertices, and these batches are disjoint. Thus $cr \leqslant v$, i.e. $c \leqslant v/r$, as required. □

4.3 On random edge-signings of fixed bicycle-free base graphs

In this section we will prove Theorem 4.1.4. In fact, we will prove the following refined version:

Theorem 4.3.1. *Let $G = (V, E)$ be an arbitrary d-regular n-vertex graph, where $d \leqslant \mathrm{polylog} n$. Assume that G is bicycle-free at radius $r \gg (\log\log n)^2$. Then for $\boldsymbol{G}$ a uniformly random edge-signing of G, except with probability at most n^{-100} the non-backtracking matrix $\boldsymbol{B}$ of $\boldsymbol{G}$ satisfies the spectral radius bound*

$$\rho(\boldsymbol{B}) \leqslant \sqrt{d-1} \cdot \left(1 + O\left(\frac{(\log\log n)^2}{r}\right)\right),$$

and hence (by Corollary 3.2.5) the signed adjacency matrix $\boldsymbol{A}$ of $\boldsymbol{G}$ satisfies the bound

$$\rho(\boldsymbol{A}) \leqslant 2\sqrt{d-1} \cdot \left(1 + O\left(\frac{(\log\log n)^4}{r^2}\right)\right).$$

Furthermore, let $C = C(n)$ satisfy $1 \leqslant C \leqslant \mathrm{polylog} n$ and suppose we merely assume that the random edge-signs are (δ, k)-wise uniform for $\delta \leqslant n^{-O(C \log d)}$ and $k \geqslant 2C \log n$. Then the above bounds continue to hold, with an additional additive $O(\sqrt{d}/C)$ in the $\rho(\boldsymbol{B})$ bound and $O(\sqrt{d}/C^2)$ in the $\rho(\boldsymbol{A})$ bound.

As in [Fri08, Bor19], the proof of Theorem 4.3.1 will use the Trace Method. In preparation for this, we make some definitions:

Definition 4.3.2 (Hikes)**.** Let $G = (V, E)$ be an undirected graph. For $\ell \in \mathbb{N}$, we define an *ℓ-hike* $\mathcal{H}$ to be a closed walk in G of exactly 2ℓ steps (directed edges) which is non-backtracking except possibly between the ℓth and $(\ell+1)$th step. Given an edge-signing $w : E \to \{\pm 1\}$ we write $w(\mathcal{H})$ for the product of the edge-signs that $\mathcal{H}$ traverses, counted with multiplicity. Finally, we call a hike *even* (respectively, *singleton-free*) if each undirected edge traversed by $\mathcal{H}$ is traversed an even number of times (respectively, at least twice).

A straightforward use of the Trace Method will now imply:

Proposition 4.3.3. *Let $\ell \in \mathbb{N}^+$ and define $\boldsymbol{T} = \operatorname{tr}\big(\boldsymbol{B}^\ell(\boldsymbol{B}^\top)^\ell\big)$ (which is an upper bound on $\rho(\boldsymbol{B})^{2\ell}$). Then for a uniformly random edge-signing $\boldsymbol{w} : E \to \{\pm 1\}$,*

$$\mathbf{E}[\boldsymbol{T}] \leqslant d^2 \cdot \#\{\text{even } (\ell-1)\text{-hikes } \mathcal{H} \text{ in } G\} \leqslant d^2 \cdot \#\{\text{singleton-free } (\ell-1)\text{-hikes } \mathcal{H} \text{ in } G\}.$$

Furthermore, if $\boldsymbol{w}$ is merely $(\delta, 2\ell)$-wise uniform, the bound holds up to an additive $\delta n d^{2\ell+2}$.

Proof. We have

$$\boldsymbol{T} = \sum_{\vec{e}_0,\vec{e}_1,\ldots,\vec{e}_{2\ell-1},\vec{e}_{2\ell}=\vec{e}_0 \in \vec{E}} \boldsymbol{B}_{\vec{e}_0,\vec{e}_1}\boldsymbol{B}_{\vec{e}_1,\vec{e}_2}\cdots\boldsymbol{B}_{\vec{e}_{\ell-1},\vec{e}_\ell}\boldsymbol{B}_{\vec{e}_{\ell+1},\vec{e}_\ell}\boldsymbol{B}_{\vec{e}_{\ell+2},\vec{e}_{\ell+1}}\cdots\boldsymbol{B}_{\vec{e}_{2\ell},\vec{e}_{2\ell-1}}. \tag{4.1}$$

Recalling the definition of $\boldsymbol{B}$, one immediately sees that $\boldsymbol{T}$ is "something like" the sum of $\boldsymbol{w}(\mathcal{H})$ over all ℓ-hikes in G. But being careful, one sees we precisely have the following:

> $\boldsymbol{T}$ is equal to the sum of $\boldsymbol{w}(\mathcal{H})$ over all "special" $(\ell+1)$-hikes in G, where we call an $(\ell+1)$-hike *special* if its $(\ell+2)$th step is the reverse of its $(\ell+1)$th step, and the last step is the reverse of the first step.[2]

Next, we employ the following easy fact:

Fact 4.3.4. *If $\boldsymbol{w} : E \to \{\pm 1\}$ is a fully uniformly random edge-signing, then $\mathbf{E}[\boldsymbol{w}(\mathcal{H})]$ will be 1 if $\mathcal{H}$ is an even hike, and will be 0 otherwise.*

Thus

$$\mathbf{E}_{\boldsymbol{w}:E\to\{\pm1\}}[\boldsymbol{T}] = \#\{\text{even, special } (\ell+1)\text{-hikes } \mathcal{H} \text{ in } G\}. \tag{4.2}$$

Since a special $(\ell+1)$-hike involves at most 2ℓ undirected edges, a crude upper bound on the number of all special $(\ell+1)$-hikes in G is $nd^{2\ell}$. Thus for an edge-signing $\boldsymbol{w}$ that is merely $(\delta, 2\ell)$-wise uniform, Equation (4.2) holds up to an additive $\delta n d^{2\ell}$. Finally, every even special $(\ell+1)$-hike $\mathcal{H}$ can be formed from an even $(\ell-1)$-hike $\mathcal{H}'$ by: (i) attaching a step and its reverse to the beginning/end of $\mathcal{H}$; (ii) attaching a step and its reverse to the midpoint of $\mathcal{H}$. As there are at most $(d-1)^2 \leqslant d^2$ choices for how to perform (i) and (ii), the inequality in the proposition's statement follows. □

At this point, edge-signs are out of the way and we are reduced to counting singleton-free hikes. In aid of this, we define the following quantities:

Definition 4.3.5. Given an $(\ell-1)$-hike $\mathcal{H}$ in a graph G, we write $G_{\mathcal{H}} = (V_{\mathcal{H}}, E_{\mathcal{H}})$ for the subgraph of G formed by the union of the edges visited by $\mathcal{H}$. We think of $G_{\mathcal{H}}$ as being "revealed" as the $2(\ell-1)$ steps of $\mathcal{H}$ are taken in order. We classify each step of $\mathcal{H}$ as either *stale*, *fresh*, or *boundary*. If a step of $\mathcal{H}$ traverses a previously-explored edge in $G_{\mathcal{H}}$ (in either direction), we call the step *stale*; otherwise, if it steps to a previously-unvisited vertex, we call the step *fresh*; otherwise, we call it *boundary*. For the purposes of this definition, at the beginning of $\mathcal{H}$ the initial vertex is considered to be "previously visited".

[2]The astute reader will note that the sign of the first/last edge in $\mathcal{H}$ is never counted in Equation (4.1); however it is okay to count it twice, as $\boldsymbol{w}(\mathcal{H})$ does, since $(\pm 1)^2 = 1$.

We now put bounds on the different kinds of steps. For the fresh steps, we only need the singleton-free property:

Proposition 4.3.6. *In a singleton-free $(\ell-1)$-hike, at least half of all steps must be stale. Thus there are fewer than ℓ fresh steps.*

For the boundary steps of $\mathcal{H}$, it is easy to see that there are exactly $\mathrm{exc}(G_{\mathcal{H}})+1$ of them.

Thus we can bound them using only the bicycle-free property. Together with the simple bound $|V_{\mathcal{H}}| \leqslant 2\ell$, Theorem 4.2.2 implies

Proposition 4.3.7. *If $\mathcal{H}$ is an $(\ell-1)$-hike in a graph G which is bicycle-free at radius $r \geqslant 10\ln(2\ell)$, then $\mathcal{H}$ has at most $O(\frac{\log \ell}{r})\cdot \ell$ boundary steps.*

Finally, to handle the stale steps we group them into "stretches".

Proposition 4.3.8. *In an $(\ell-1)$-hike $\mathcal{H}$, the stale steps may be partitioned into at most $O(\frac{\log \ell}{r})\cdot \ell$ stretches of consecutive stale steps, each stretch having length at most r, and none straddling the "turnaround" at step ℓ.*

Proof. We begin by partitioning the stale steps into maximal contiguous stretches. It is easy to see that each of these must be preceded in $\mathcal{H}$ by a boundary step (with a single possible exception of the "turnaround" at step ℓ). Thus Proposition 4.3.7 implies that there are at most $O(\frac{\log \ell}{r})\cdot \ell$ maximal stretches of stale steps. If a maximal stretch straddles the turnaround, we can split it in two. Finally, if necessary we now subdivide the stretches into length at most r. Since there are fewer than 2ℓ stale steps, this subdivision can be done without increasing the number of stretches by more than $2\ell/r \leqslant O(\frac{\log \ell}{r})\cdot \ell$. □

We may now make our final estimate:

Theorem 4.3.9. *In a d-regular graph G that is bicycle-free at radius $r \geqslant 10\ln(2\ell)$, the number of singleton-free $(\ell-1)$-hikes $\mathcal{H}$ is at most $O(\ell^3 n)\cdot(d-1)^{\ell}\cdot(dr\ell)^{O(\frac{\log \ell}{r})\cdot\ell}$.*

Proof. Following [Bor19], we use an encoding argument. To each $\mathcal{H}$ we associate a string $\textsc{struct}(\mathcal{H})$ over the alphabet $\{\textsc{f},\textsc{b},\textsc{s}\}$, where we replace each fresh step with an F, each boundary step with a B, and each stale **stretch** with an S. Our goal will be to show:

Claim 4.3.10. For any string σ with c_f, c_b, c_s occurrences of F, B, S (respectively), there are no more than $2n\cdot(d-1)^{c_f+c_b}\cdot(2r\ell)^{c_s}$ singleton-free $(\ell-1)$-hikes $\mathcal{H}$ with $\textsc{struct}(\mathcal{H})=\sigma$.

Let us complete the proof of the theorem assuming this claim. By Propositions 4.3.6 to 4.3.8, we have the bounds

$$c_f < \ell, \qquad c_b, c_s < m := O(\tfrac{\log \ell}{r})\cdot \ell.$$

Crudely, there are at most $O(\ell^3)$ possibilities for the triple (c_f, c_b, c_s). Also, the following two quantities are increasing in c_f, c_b, c_s:

$$2n\cdot(d-1)^{c_f+c_b}\cdot(2r\ell)^{c_s}, \qquad \Sigma_{c_f,c_b,c_s} := \#\text{ strings of } c_f \text{ F's, } c_b \text{ B's, } c_s \text{ S's.}$$

Thus we can upper-bound the number of all singleton-free $(\ell-1)$-hikes by

$$O(\ell^3 n)\cdot(d-1)^{\ell+m}\cdot(2r\ell)^m\cdot\Sigma_{\ell,m,m} \leqslant O(\ell^3 n)\cdot(d-1)^{\ell}\cdot(dr\ell)^{O(m)},$$

as needed, where we used the simple bound $\Sigma_{\ell,m,m}\leqslant \ell^{O(m)}$.

It remains to prove the claim. Let σ be as given. We may recover all possible associated $\mathcal{H}$, in a vertex-by-vertex fashion, by first specifying the initial vertex (n choices) and then proceeding through the symbols of σ in order. If we are at an F or a B symbol, we can recover the next vertex by specifying one of $d-1$ neighbors of the current vertex; there are only $d-1$ possibilities, since $\mathcal{H}$ is non-backtracking. (Exception: there are d choices at the very beginning of the hike; we compensated for this with the factor $2 > \frac{d}{d-1}$.) To complete the proof of the claim, we need to show that for each stale stretch, there are at most $2r\ell$ possibilities. Recall that a stale stretch beginning from a vertex v consists of walking in non-backtracking fashion for at most r steps over the previously seen portion K of $G_{\mathcal{H}}$. This subgraph K has at most 2ℓ vertices, and by the bicycle-free property, this walk is confined to a subgraph of K that is at most unicyclic. It is easy to see this walk is determined by specifying its final vertex (at most 2ℓ possibilities), the number of times the cycle in v's distance-r neighborhood (should it exist) is traversed (fewer than $r/2$ possibilities), and the direction in which the cycle is traversed (2 possibilities). Thus indeed each stale stretch can be completely determined by specifying one of at most $2\ell \cdot (r/2) \cdot 2 = 2r\ell$ possibilities. □

Combining this with Proposition 4.3.3 now yields:

Corollary 4.3.11. *Let $G = (V, E)$ be an arbitrary d-regular n-vertex graph. Assume that G is bicycle-free at radius r. Let $\ell \in \mathbb{N}^+$ and $0 < \eta < 1$ be parameters. Then for $\boldsymbol{G}$ a uniformly random edge-signing of G, except with probability at most η the non-backtracking matrix $\boldsymbol{B}$ of $\boldsymbol{G}$ has spectral radius bound*

$$\rho(\boldsymbol{B}) \leqslant \sqrt{d-1} \cdot (1 + O(\varepsilon_1) + O(\varepsilon_2)), \tag{4.3}$$

where

$$\varepsilon_1 \coloneqq \frac{\log(n/\eta)}{\ell}, \qquad \varepsilon_2 \coloneqq \frac{\log(d\ell)\log(\ell)}{r},$$

provided $\varepsilon_1, \varepsilon_2 \leqslant 1$.

Furthermore, if the random edge-signs of $\boldsymbol{G}$ are merely $(\delta, 2\ell)$-wise uniform, the bound holds up to an additional additive $(\delta n/\eta)^{\frac{1}{2\ell}} \cdot O(d)$.

Proof. We have obtained that, for a uniformly random edge-signing $\boldsymbol{w} : E \to \{\pm 1\}$,

$$\mathbf{E}[\boldsymbol{T}] \leqslant O(d^2\ell^3 n) \cdot (d-1)^\ell \cdot (dr\ell)^{O(\frac{\log \ell}{r})\cdot \ell}.$$

Note that $r \lesssim \log_{d-1} n$ always holds, and hence we must have $\ell \leqslant n$ (else $\varepsilon_2 > 1$). Also we must have $\ell \geqslant \log n$ (else $\varepsilon_1 > 1$). Thus we may coarsen $O(d^2\ell^3 n)$ in the above to $O(n^5)$, and coarsen $(dr\ell)^{O(\cdot)}$ to $(d\ell)^{O(\cdot)}$. Now since $\boldsymbol{T}$ is a nonnegative random variable, Markov's inequality implies that except with probability at most η,

$$\boldsymbol{T} \leqslant O(n^5/\eta) \cdot (d-1)^\ell \cdot (d\ell)^{O(\frac{\log \ell}{r})\cdot \ell},$$

and hence

$$\rho(\boldsymbol{B}) \leqslant \boldsymbol{T}^{\frac{1}{2\ell}} \leqslant O(n^5/\eta)^{\frac{1}{2\ell}} \cdot \sqrt{d-1} \cdot (d\ell)^{O(\frac{\log \ell}{r})},$$

which directly implies Equation (4.3).

Finally, in the $(\delta, 2\ell)$-wise uniform case, we get an additional additive $\delta n d^{2\ell+2}$ in the bound on $\mathbf{E}[\boldsymbol{T}]$; this gets a factor of $1/\eta$ after the application of Markov, and becomes $(\delta n/\eta)^{\frac{1}{2\ell}} \cdot O(d)$ after taking 2ℓth roots. □

Finally, the reader may verify that Theorem 4.3.1 follows from Corollary 4.3.11 in the fully uniform case by taking $\ell = \Theta(r \log(n)/\log\log n)$, and in the derandomized case by taking $\ell = \Theta(C \log(n/\eta))$.

Remark 4.3.12. Alternatively, by taking $\eta = \exp(-\exp(r^{.49}))$ and $\ell = \exp(r^{.49})$ in Corollary 4.3.11, we may conclude that $\rho(\boldsymbol{B}) \leqslant \sqrt{d-1} \cdot (1 + o_r(1))$ holds in the fully uniform case except with probability at most $\exp(-\exp(r^{.49}))$.

4.4 Weakly derandomizing Bordenave's theorem for random lifts

Bordenave [Bor19] also confirmed Theorem 1.1.16 in the case that $\boldsymbol{G}$ is a random n-lift of any fixed d-regular Ramanujan base graph $\underline{G}$. The simplest case is $\underline{G} = K_{d+1}$, the complete graph on $d+1$ vertices. This gives a way to randomly construct arbitrarily large d-regular near-Ramanujan graphs that are always simple. In particular, Bordenave proves:

Theorem 4.4.1. *Fix any $d \geqslant 3$ an $\epsilon > 0$ and let $\boldsymbol{G}$ be a random n-lift of K_{d+1}. Then*

$$\mathbf{Pr}[\lambda(G) \leqslant 2\sqrt{d-1} + \epsilon] \geqslant 1 - o_n(1).$$

In this section we give a weak derandomization of Bordenave's proof of Theorem 4.4.1, using "off-the-shelf" tools; the derandomization is "weak" in the sense that it yields a $\mathrm{quasipoly}(n)$-time deterministic construction. Specifically, we will show that the conclusion of Theorem 4.4.1 holds even for the "almost k-wise uniform" lift model, $k = O(\log n)$.

Definition 4.4.2 ((δ, k)-wise uniform lift model)**.** When the permutations $\boldsymbol{\pi}_{uv}$ are each not uniformly random but are merely (δ, k)-wise uniform, we will say that $\boldsymbol{G}_n$ is drawn from the (δ, k)-wise uniform lift model, or equivalently, $\boldsymbol{G}_n$ is a random (δ, k)-wise uniform lift of G_0.

We will not fully recap Bordenave's proof of Theorem 4.4.1 in this work, although the reader unfamiliar with it will get some insight knowing that our proof of Theorem 4.3.1 is modeled on it. Bordenave employs two twists on the Trace Method to show that a random n-lift $\boldsymbol{G}$ of K_{d+1} has spectral radius at most $2\sqrt{d-1} + \varepsilon$ (when the trivial eigenvalue of d is ignored). The less important (but still challenging) twist involves replacing the non-backtracking matrix $\boldsymbol{B}$ by a centered variant, $\underline{\boldsymbol{B}}$, that enables one to ignore the trivial eigenvalue. The more conceptually important twist comes from the fact, originally recognized by Friedman, that even after passing to $\underline{\boldsymbol{B}}$, the Trace Method still fails. The reason, in brief, is as follows: A successful use of the Trace Method would have to consider walks of length ℓ for ℓ at least a large multiple of $\log n$, in order to overcome the factor of n arising from the n different walk starting points (cf. the error term ε_1 just after Equation (4.3)). But for walks of this long length, one can show that the expected trace of $\underline{\boldsymbol{B}}^{\ell}(\underline{\boldsymbol{B}}^{\top})^{\ell}$ is simply too large — much larger than the target $\mathrm{poly}(n) \cdot (d-1)^{\ell}$ needed to get the "correct" final bound.

However, as first demonstrated by Friedman in the case of random d-regular graphs, the expectation is too large only because of certain low-probability events.

Bordenave's way of handling things is to show that: (i) a random n-lift $\boldsymbol{G}$ of a d-regular graph is, with high probability, bicycle-free at large radius r; (ii) when $\boldsymbol{G}$ is so bicycle-free, the rth power of its non-backtracking matrix, $\boldsymbol{B}^r$, coincides with a certain "bicycle-discarding" variant $\boldsymbol{B}^{(r)}$; (iii) the usual Trace Method *can* be successfully applied to $\boldsymbol{B}^{(r)}$; i.e., the expected trace of powers of $\underline{\boldsymbol{B}}^{(r)}$ *is* suitably small.

Thus our weak derandomization of Bordenave's proof has two ingredients, corresponding to (i) and (iii) above. In Section 4.4.1 we derandomize a standard proof that a random n-lift of a d-regular graph is bicycle-free at large radius. In Section 4.4.2 we examine the key probabilistic ingredient in Bordenave's use of the Trace Method, [Bor19, Prop. 28], which encapsulates the fact that for a centered version $\underline{\boldsymbol{M}}$ of the configuration model matching matrix, the random variables $\underline{\boldsymbol{M}}_{(v,i),(v',i')}$ are close to k-wise independent for $k \ll \sqrt{dn}$.

4.4.1 Derandomizing Bicycle-freeness

The following relatively straightforward fact about d-regular n-lifted graphs is crucial for Bordenave's proof: with high probability they are bicycle-free at radius r, provided $r \lesssim c \log_{d-1} n$ for some constant $c < 1/4$. This fact is proved for completeness by Bordenave [Bor19, Lem. 27]. We would like a derandomized version of this fact for the k-wise uniform configuration model, $k = O(r)$. This motivates looking for a moments-based proof, such as the one suggested by Wormald [Wor99, Lem. 2.7] and carried out for Erdős–Rényi $\mathcal{G}(n, m)$ graphs in [JŁR00, Thm. 5.5]. The essential point will be that minimal witnesses to failure have only $O(r)$ edges.

Definition 4.4.3 (Minimal bicycle)**.** We say a connected multigraph is a *minimal bicycle* if it is bicyclic but has no proper subgraph that is bicyclic. It is easy to see (cf. [JŁR00, Proof of Thm. 5.5]) that any minimal bicycle is either a "handcuffs graph" (two cycles joined by a path), a "figure-eight graph" (two cycles attached at a vertex), or a "theta graph" (a cycle with a "diagonal").

We now prove:

Proposition 4.4.4. *Fix $d \geqslant 3$ and $k \geqslant 1$. Let $\boldsymbol{G}$ be an n-lift of K_{d+1} drawn from the $2k$-wise uniform lift model. Then $\boldsymbol{G}$ is bicycle-free at radius $k/4$, except wth probability at most $O(k^3(d+1)^k/n)$. In particular, when G_0 is K_{d+1}, the failure probability is at most $1/n^{.99}$ provided $k < c \log_{d-1} n$ for a certain universal constant $0 < c < 1/4$.*

By Theorem 3.4.5, this remains true if the permutation is $(\delta, 2k)$-wise uniform, $\delta \leqslant 1/(n^{8k+2})$.

Proof. Fix a minimal bicycle H with h vertices and hence $h+1$ edges, where $h < k$. Let the random variable $\boldsymbol{X}_H$ denote the number of times that H appears in $\boldsymbol{G}$. This is a polynomial of degree at most $h+1 \leqslant k$ in the entries of $\boldsymbol{G}$'s adjacency matrix and hence a polynomial of degree at most $2k$ in the permutation indicators $1[\boldsymbol{\pi}_{uv}(j) = i]$. Thus to compute $\mathbf{E}[\boldsymbol{X}_H]$ we may assume $\boldsymbol{G}$ is a uniformly random n-lift of G_0. In this case,

$$\mathbf{E}[\boldsymbol{X}_H] = \sum_{S \in \left\{\text{subgraphs of } K_{(d+1)\cdot n} \text{ isormorphic to } H\right\}} \mathbf{Pr}[S \text{ occurs as subgraph of } \boldsymbol{G}]$$

$$\leqslant \#\text{ of occurrences of } H \text{ in } K_{(d+1)\cdot n} \times \frac{1}{n(n-1)\cdots(n-h)}$$
$$\leqslant \frac{(n\cdot|V_0|)^h}{n(n-1)\cdots(n-h)}$$
$$\leqslant O\left(\frac{|V_0|^h}{n}\right)$$

Finally, it is easy to see that, up to isomorphism, the number of minimal bicycles with fewer than k vertices is at most $O(k^3)$. Thus by Markov's inequality we conclude that the probability of having any minimal bicycle on fewer than k vertices is at most $O(k^3(d+1)^k/n)$. The proposition now follows since any bicyclic radius-$k/4$ vertex neighborhood in $\boldsymbol{G}$ must contain a minimal bicycle with fewer than k vertices. (The "worst case" is a figure-eight graph.) □

4.4.2 Bound on the modified trace

In this section we examine the last place in Bordenave's argument that uses randomness of the underlying graph $\boldsymbol{G}$; namely, [Bor19, Prop. 28], an upper bound on a certain moment arising in his use of the Trace Method. Unfortunately, this proposition is not as self-contained as the one covered in Section 4.4.1. Rather than trying to give a complete summary of how Bordenave's argument works, we will proceed in a "black-box" fashion, only giving the bare minimum needed to verify derandomizability. We refer the reader to [Bor19] for the complete picture.

Here is the key probabilistic proposition (which can be viewed as a far more sophisticated version of Fact 4.3.4):

Proposition 4.4.5 ([Bor19, Prop. 28])**.** *Let $\boldsymbol{G}$ be an n-lift of K_{d+1}, and for $e \in \vec{E}(K_{d+1})$ let $\boldsymbol{M}_e$ be the uniformly random permutation matrix corresponding to permutation $\boldsymbol{\pi}_e$ as in the lift model in Definition 3.3.1. Also let $\underline{\boldsymbol{M}}_e$ be the matrix obtained from $\boldsymbol{M}_e$ by subtracting $\frac{1}{n}$ from each entry. Then for any length-k sequence of tuples $(e_1,(i_1,j_1)),\ldots,(e_k,(i_k,j_k))$ where each e_i is a directed edge in $\vec{E}_0$ and (i_t,j_t) is in $[n]\times[n]$ with $1 \leqslant k \leqslant \sqrt{n}$, and for any $0 \leqslant k_0 \leqslant k$, we have*

$$\left|\mathbf{E}\left[\prod_{t=1}^{k_0}\underline{\boldsymbol{M}}_{e_t}[i_t,j_t]\prod_{t=k_0+1}^{k}\boldsymbol{M}_{e_t}[i_t,j_t]\right]\right| \leqslant O\Big(2^b\cdot(\tfrac{1}{n})^a\cdot(\tfrac{3k}{\sqrt{n}})^{a_1}\Big). \tag{4.4}$$

Here a, b, and a_1 on the right-hand side of Equation (4.4) are certain quantities relating to the combinatorial properties of the sequence $(e_1,(i_1,j_1)),\ldots,(e_k,(i_k,j_k))$. We omit these definitions here, as they won't be relevant for us. The relevant point of the above proposition is that there is *some* anonymous quantity bounding the left hand side of Equation (4.4).

Note that when $\boldsymbol{M}_e$ is formed from a random permutation $\boldsymbol{\pi}_e$ on $[n]$ as in Definition 3.3.1, each entry $\boldsymbol{M}_e[i,j]$ is a polynomial of degree 1 in the indicators $1[\boldsymbol{\pi}_e(i)=j]$. It follows that the quantity inside the expectation in Equation (4.4) is a polynomial of degree at most k in these indicators. We conclude:

Corollary 4.4.6. *Let $\boldsymbol{G}$ be drawn by taking a random lift of G_0 using $2k$-wise uniform permutations, and write $\boldsymbol{M}$ for the matching matrix inducing $\boldsymbol{G}$. Then Equation* (4.4) *continues to hold.*

With Proposition 4.4.5 in hand, Bordenave does some intricate — but entirely non-probabilistic — path-counting to complete his use of the Trace Method. (This is like a much more sophisticated version of the part of Section 4.3 beginning with Definition 4.3.2.) This part of his proof involves considering paths of length $2\ell m$, where "ℓ" and "m" are parameters he selects (with ℓ being at least the bicycle-free radius, and m being large enough so that $\ell m \gg \log n$). The crucial observation for us is that Bordenave *only* employs Proposition 4.4.5 with its parameter "k" set to $2\ell m$.

Bordenave directly sets $\ell = \Theta(\log_{d-1}(n))$ and $m = \Theta(\log(n)/\log\log(n))$ to obtain best parameters, but we will work more generally, since we may be interested in minimizing $k = 2\ell m$ to save on random bits. Carefully examining [Bor19, Proofs of Prop. 29, 33], one may extract the below proposition. The random matrices $\underline{\boldsymbol{B}}^{(\ell)}$ and $\boldsymbol{R}_1^{(\ell)}, \dots, \boldsymbol{R}_\ell^{(\ell)}$ mentioned in it are derived from the randomness of the lift model; again, see [Bor19] for details.

Proposition 4.4.7. *Assuming d, ℓ, m satisfy $\ell m \ll \sqrt{dn}$ and $\mathrm{poly}(d\ell m)^m \ll n$,*

$$\mathbf{E}\left[\|\underline{\boldsymbol{B}}^{(\ell)}\|^{2m}\right] \leqslant n \cdot \mathrm{poly}(d\ell m)^m \cdot (d-1)^{\ell m}, \qquad \mathbf{E}\left[\sum_{i=1}^{\ell} \|\underline{\boldsymbol{R}}_i^{(\ell)}\|^{2m}\right] \leqslant \mathrm{poly}(d\ell m)^m \cdot (d-1)^{2\ell m},$$

Furthermore, this only relies on Equation (4.4) *with $k = 2\ell m$, and therefore by Corollary 4.4.6 it continues to hold even in the $4\ell m$-wise uniform lift model. Thus in this model, Markov's inequality implies that except with probability at most n^{-100},*

$$\|\underline{\boldsymbol{B}}^{(\ell)}\| \leqslant \mathrm{poly}(n)^{\frac{1}{2m}} \cdot (d-1)^{\frac{\ell}{2}}, \qquad \sum_{i=1}^{\ell} \|\underline{\boldsymbol{R}}_i^{(\ell)}\| \leqslant \mathrm{poly}(n)^{\frac{1}{2m}} \cdot (d-1)^{\ell}.$$

Finally, [Bor19, Prop. 26] is the following:

Proposition 4.4.8. *Suppose $\boldsymbol{G}$ drawn from the lift model is bicycle-free at radius ℓ. Then the largest magnitude "new" eigenvalue of the associated non-backtracking matrix $\boldsymbol{B}$ is at most*

$$\left(\|\underline{\boldsymbol{B}}^{(\ell)}\| + \frac{1}{n} \cdot \sum_{i=1}^{\ell} \|\underline{\boldsymbol{R}}_i^{(\ell)}\|\right)^{1/\ell}.$$

We can now finish the proof as Bordenave does. Using the parameter settings $\ell = c \log_{d-1} n$ and $m = (C/c)\log(d-1)/\sqrt{\epsilon}$ where c and C is a large enough universal constant, and combining Corollary 3.2.5 and Propositions 4.4.4, 4.4.7 and 4.4.8, we get the following:

Theorem 4.4.9. *Fix any $d \geqslant 3$ and $\epsilon > 0$, and let $k \geqslant C \log n/\sqrt{\varepsilon}$. Let $\boldsymbol{G}_n$ be a random k-wise uniform lift of K_{d+1}. Then except with probability at most $1/n^{.99}$, the following hold:*

- *$\boldsymbol{G}_n$ is bicycle-free at radius $c \log_{d-1} n$;*
- *$\lambda(\boldsymbol{G}_n) \leqslant 2\sqrt{d-1} + \epsilon$.*

Finally, by Theorem 3.4.5, these statements remain true for (δ, k)-wise uniform lifts, where $\delta < 1/n^{8k+1}$.

4.5 Explicit near-Ramanujan graphs

With the tools developed in Section 4.3 and Section 4.4 we are now ready to establish our explicit near-Ramanujan graph constructions. For ease of reading, in this section we will merely prove

a weaker version of Theorem 4.1.1, the deterministic polynomial-time ("weakly explicit") construction, with d and ε assumed to be constant. We leave the slightly more technical proof of the "probabilistically strongly explicit" construction (Theorem 4.1.1), with worked out dependence on $d = d(n)$ and $\varepsilon = \varepsilon(n)$, for Section 4.6.

Recall we want to show there is a deterministic algorithm that on input $N, d \geqslant 3$ and $\varepsilon > 0$, outputs in $\mathrm{poly}(N)$-time a d-regular graph G on $N' \sim N$ vertices with $\lambda(G) \leqslant 2\sqrt{d-1} + \varepsilon$.

Before getting into the details, we recap the construction as outlined in Section 4.1.2:

1. Using Theorem 4.4.9 we construct a d-regular simple graph G_0 on some "small" number of vertices $n_0 = n_0(N)$, which is bicycle-free at radius $\Omega(\log n_0)$ and has $\lambda(G_0) \leqslant 2\sqrt{d-1} + \epsilon$. The quantity n_0 should satisfy

$$2^{\omega((\log\log N)^2)} \leqslant n_0 \leqslant 2^{O(\sqrt{\log N})},$$

   the left inequality so that G_0 is sufficiently bicycle-free for Step 2 below, and the right inequality so that G_0 is constructible in deterministic $\mathrm{poly}(N)$ time. We have a wide range of allowable possibilities here; for concreteness we will take n_0 near the upper limit to allow for slightly better dependence on non-constant d, ε in Section 4.6.

2. Next we repeatedly use Theorem 4.3.1 (roughly $\log(N/n_0) \sim \log N$ times) to double the number of vertices in our construction from Step 1, while keeping $\lambda \leqslant 2\sqrt{d-1} + \epsilon$ and also retaining that the graph is bicycle-free at radius $\Omega(\log n_0)$ (Proposition 4.2.1). Importantly, since Theorem 4.3.1 is a high-probability result, we will be able to reuse the seed for each of the $\log N$ pseudorandom edge-signings.

Step 1 details. Here the algorithm will select n_0 to be an even integer on the order of $2^{\Theta(\sqrt{\log N})}$. Theorem 4.4.9 tells us that for a sufficiently large $k = O(\log n_0) = O(\sqrt{\log N})$, and for sufficiently small $\delta = n_0^{-\Theta(k)} = 1/\mathrm{poly}(N)$, a random d-regular n_0-vertex graph $\boldsymbol{G}_0$ chosen from the (δ, k)-wise uniform random-lift-of-K_{d+1} model will with high probability satisfy:

$$\boldsymbol{G}_0 \text{ is bicycle-free at radius } \Omega(\log n_0) = \Omega(\sqrt{\log N}); \qquad \lambda(\boldsymbol{G}_0) \leqslant 2\sqrt{d-1} + \varepsilon. \tag{4.5}$$

(Recall we are treating d and ε as constant here.) $\boldsymbol{G}_0$ will also be simple with $\Omega(1)$ probability in the configuration model case, and with probability 1 in the random lift case. In the former case, we need a (δ, k)-wise permutation in S_{nd}; in the latter case, we need $\binom{d+1}{2}$ independent (δ, k)-wise permutations in S_n. Either way, Theorem 3.4.4 tells us that a deterministic algorithm can enumerate all possibilities for $\boldsymbol{G}_0$ in $\mathrm{poly}(N)$ time and pick out any fixed simple one G_0 satisfying (4.5).

Step 2 details. Here the algorithm will be applying Theorem 4.3.1 some $t \sim \log_2 N$ times, starting with G_0, and each time interpreting the edge-signing produced as a 2-lift as discussed in Section 4.1.2. This produces a sequence of pseudorandom d-regular simple graphs $\boldsymbol{G}_1, \ldots, \boldsymbol{G}_t$, where $\boldsymbol{G}_i$ has $n_0 2^t$ vertices. The parameter t is chosen to be least possible such that the final number of vertices, $N' = n_0 2^t$, is as at least N. It is not hard to check that by adjusting n_0 by a factor of at most 2, we can ensure that $N'/N = 1 + o_N(1)$, where the $o_N(1)$ term is $O(1/n_0) = 1/2^{\Theta(\sqrt{\log N})}$.

For simplicity, we will use the same values for the parameters r, k, and δ in each application of Theorem 4.3.1; only the value of n will change (ranging from n_0 up to N'). We may take $r = \Omega(\sqrt{\log N})$, the bicycle-free radius from Equation (4.5) (observe that the bicycle-free radius cannot decrease for *any* 2-lift of a graph). Note that the failure probability of any single 2-lift is at most $1/2^{\Theta(\sqrt{\log N})}$, and hence a union bound tells us that the probability of *any* of the 2-lifts "failing" is low, $\frac{\log N}{2^{\Theta(\sqrt{\log N})}}$. We take the parameter "k" to be $\Theta\left(\frac{\log N}{\sqrt{\varepsilon}}\right)$ (the hidden constant sufficiently large depending on d). Finally, we take $\delta = 1/N^{\Theta(1/\sqrt{\varepsilon})}$ (again with the hidden constant sufficiently large depending on d). By plugging these parameters into Theorem 4.3.1 we conclude that with high probability, all "new" eigenvalues arising in the 2-lifted adjacency matrices $\boldsymbol{A}_1, \dots, \boldsymbol{A}_t$ are at most $2\sqrt{d-1} + \varepsilon$ in magnitude, and hence $\boldsymbol{G}_t$ is ε-near Ramanujan.

It remains to observe that with these parameter settings, using Theorem 3.4.2, a deterministic algorithm can in $\mathrm{poly}(N/\delta) = \mathrm{poly}(N)$ time do the following: First, produce a single $(\delta, 2\ell)$-wise uniform multiset of strings $Y \subseteq \{\pm 1\}^{N'd/4}$; here $N'd/4$ bits are sufficient to edge-sign/2-lift any of the graphs $\boldsymbol{G}_i$. Then, the algorithm can search Y for a "good" string $y \in Y$, meaning one with the property that using (a prefix of) it to do *each* of the t edge-signings/2-lifts yields graphs $G_1, G_2, \dots, G_t$ all of which are ε-near Ramanujan. As argued in the previous paragraph, a $1 - O\left(\frac{\log N}{2^{\Theta(\sqrt{\log N})}}\right)$ fraction of strings in Y have this property, and by Fact 4.1.2 we can check the goodness of any string y in $\mathrm{poly}(N)$ time.

This concludes the proof.

4.6 The probabilistically strongly explicit construction

We now walk through the steps of Section 4.5 giving precise parameter details along the way, and extract a probabilistically strongly explicit construction of near-Ramanujan graphs.

Assume we are given N, $3 \leqslant d \leqslant \frac{(\log N)^{1/8}}{C}$ and $\epsilon \gg \frac{(\log\log N)^4}{\log N} \cdot \sqrt{d}$ where C is the constant from the statement of Theorem 4.4.9.

Revisiting Step 1. Choose parameters as follows: $\alpha = 1/\sqrt{\binom{d+1}{2}}$; n_0 as the largest multiple of $d+1$ smaller than $2^{\alpha\sqrt{\log N}}$; $k = C\alpha\sqrt{\log N} \cdot d^{1/4}/\sqrt{\epsilon}$ (which is $\approx \log n_0$); and $\delta = 1/N^{8k+1}$. Recall that the key result used in this step is that by Theorem 4.4.9, $\boldsymbol{G}_0$ drawn from the n_0-vertex (δ, k)-wise random-lift-of-K_{d+1} model is a simple graph that with high probability satisfies:

$$\boldsymbol{G}_0 \text{ is bicycle-free at radius } \Omega\left(\frac{\alpha\sqrt{\log N}}{\log(d-1)}\right); \qquad \lambda(\boldsymbol{G}_0) \leqslant 2\sqrt{d-1} + \varepsilon. \tag{4.6}$$

As an upshot of Theorem 3.4.4, $\boldsymbol{G}_0$ can be sampled using $\boldsymbol{s}$, a uniform binary string of length $O\left(\frac{\log N \cdot d^{1/4}}{\sqrt{\epsilon}}\right)$ as a seed. In particular, $\boldsymbol{s}$ is divided into $\binom{d+1}{2}$ disjoint substrings $\boldsymbol{s}_{e_1}, \dots, \boldsymbol{s}_{e_{\binom{d+1}{2}}}$ each of length $\ell_1 = O\left(\frac{\alpha^2 \log N \cdot d^{1/4}}{\sqrt{\epsilon}}\right)$ indexed by edges of K_{d+1}; the (δ, k)-wise uniform permutation $\boldsymbol{\pi}_{uv}$ corresponding to edge (u, v) is taken to be the $\boldsymbol{s}_{uv}$th permutation in the multiset of

permutations Π from the statement of Theorem 3.4.4. Additionally, given $\boldsymbol{s}$ and a vertex $(u, i) \in V(\boldsymbol{G}_0)$, it is possible to return a list of its neighbors in time $T_1 = O\Big(d \cdot \mathrm{poly}\Big(\frac{\alpha^2 \log N \cdot d^{1/4}}{\sqrt{\epsilon}}\Big)\Big)$.

Revisiting Step 2. Let $t = \left\lceil \log\left(\frac{N}{n_0}\right) \right\rceil$; let β be a large enough constant; let $k = \frac{2\beta d^{1/4}}{\sqrt{\epsilon}} \log N$; and let $\delta = N^{-O(\beta d^{1/4} \log d/\sqrt{\epsilon})}$. The main result used in Step 2 is that from Theorem 4.3.1 the graphs $\boldsymbol{G}_1, \ldots, \boldsymbol{G}_t$ where $\boldsymbol{G}_i$ is obtained via a 2-lift of $\boldsymbol{G}_{i-1}$ induced by a (δ, k)-wise uniform signing have their nontrivial eigenvalues bounded by $2\sqrt{d-1} + \epsilon$ in magnitude, except with probability $O(t/n_0^{100})$. From Theorem 3.4.2, a (δ, k)-wise uniform signing of any $\boldsymbol{G}_i$ can be obtained by first sampling a random binary string $\boldsymbol{s}'$ of length $\ell_2 = O\left(\frac{d^{1/4} \log d \cdot \log N}{\sqrt{\epsilon}}\right)$ and choosing the $\boldsymbol{s}$th string in the multiset of signings Y from the theorem statement. In fact, given $\boldsymbol{s}'$ and edge $e \in \boldsymbol{G}_i$ one can also output the sign assigned to edge e in time $T_2 = \mathrm{poly}\left(\beta d^{1/4} \log d \log N/\sqrt{\epsilon}\right)$. Finally, the bound of $O(t/n_0^{100})$ on the probability that $\boldsymbol{G}_t$ is not ϵ-near Ramanujan holds even if we use (a prefix of) the same seed $\boldsymbol{s}$ to perform each of the 2-lifts. Note that $t < \log N$ and $n_0 \geqslant 2^{(\log N)^{1/4}}$ and hence the failure probability is $o_N(1)$.

Probabilistically strongly explicit near-Ramanujan graphs. Given a uniform binary string $\boldsymbol{s}$ of length $\ell_1 + \ell_2$ as a random seed, call the substring given by the first ℓ_1 bits $\boldsymbol{s}_1$ and the substring given by the next ℓ_2 bits $\boldsymbol{s}_2$. Let $\boldsymbol{G}_0$ be sampled from $\boldsymbol{s}_1$ as described in Step 1, and let $\boldsymbol{G}_t$ be the "final graph" obtained by the sequence of 2-lifts in Step 2 from $\boldsymbol{s}_2$. Each vertex in $\boldsymbol{G}_i$ can be naturally identified with a tuple $(v, a, x) \in [d] \times [n_0] \times \{0,1\}^i$. Let x be a string in $\{0,1\}^t$, let $x^{\leqslant i}$ denote its i-bit prefix. Given a vertex (v, a, x) in $\boldsymbol{G}_t$ and seeds $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, we describe an algorithm to output a list of its d neighbors in $\widetilde{O}(T_1 + dT_2)$-time where the $\widetilde{O}(\cdot)$ hides factors of $\mathrm{polylog}N$. From Step 1, we know that there is an T_1-time algorithm to output a list of d neighbors of $(v, a, x^{\leqslant 0})$ in $\boldsymbol{G}_0$.

Next, given a list of neighbors of $(v, a, x^{\leqslant i-1})$ in $\boldsymbol{G}_{i-1}$ it is possible to output a list of neighbors of $(v, a, x^{\leqslant i})$ in $\boldsymbol{G}_i$ in $\widetilde{O}(dT_2)$-time in the following way. Let (w, b, y) be a neighbor of $(v, a, x^{\leqslant i-1})$. Then exactly one of $(w, b, y \wedge 0)$ and $(w, b, y \wedge 1)$ is a neighbor of $(v, a, x^{\leqslant i})$ where $\wedge$ denotes concatenation. It is possible to obtain the sign on edge $\{(v, a, x^{\leqslant i-1}), (w, b, y)\}$ in the 2-lift from $\boldsymbol{G}_{i-1}$ to $\boldsymbol{G}_i$ in T_2 time from $\boldsymbol{s}_2$. If the sign is a -1, then $(w, b, y \wedge (1 - x_i))$ is a neighbor of $(v, a, x^{\leqslant i})$; otherwise $(w, b, y \wedge x_i)$ is a neighbor. Thus, in $\widetilde{O}(dT_2)$ time, we can obtain a length-d (and hence complete) list of neighbors of $(v, a, x^{\leqslant i})$.

As a result, after spending T_1 time generating a list of neighbors of $(v, a, x^{\leqslant 0})$, we can use the above routine t times to obtain a list of neighbors of (v, a, x) in $\boldsymbol{G}_t$ in $T_1 + t \cdot \widetilde{O}(dT_2) \leqslant \widetilde{O}(T_1 + dT_2)$. From the upper and lower bounds on d and ϵ, this quantity is always $O(\mathrm{polylog}N)$.

To summarize, we have an algorithm that takes in a random seed of length $O\left(\frac{d^{1/4} \log d \cdot \log N}{\sqrt{\epsilon}}\right)$ and implements the adjacency matrix of a corresponding random graph $\boldsymbol{G}$ such that:

- Given any vertex v of $\boldsymbol{G}$, its list of neighbors can be generated in $O(\mathrm{polylog}N)$ time.
- $\boldsymbol{G}$ is ϵ-near Ramanujan with probability $1 - o_N(1)$.

This yields the conclusion of Theorem 4.1.1.

Chapter 5

Additive Lifts, CSPs and Two-Eigenvalue Graphs

In joint work with Sidhanth Mohanty and Ryan O'Donnell [MOP20b], we precisely determined the SDP value of large random instances of certain kinds of constraint satisfaction problems, which are known as "two-eigenvalue 2CSPs", which we describe in this chapter. Briefly, these are CSPs where each clause can be described by a graph where each vertex represents a variable and each edge is an XOR constraint between two variables, and such that the spectrum of the adjacency matrix of the graph only contains two distinct eigenvalues. This includes multiple famous CSPs families like the NAE-3SAT, the SORT_4 and the Forrelation_k CSPs.

To establish this result we analyze the spectral expansion of a distribution of graphs that generalizes uniformly random regular graphs. To do so we generalize well known concepts like the nonbacktracking operator, the Ihara-Bass Formula, and the Friedman/Bordenave proof of Alon's Conjecture.

5.1 Background

In the theory of algorithms and complexity, the most difficult instances of a given constraint satisfaction problem (CSP) are arguably random (sparse) instances. Indeed, the assumed intractability of random CSPs underlies various cryptographic proposals for one-way functions [Gol00, JP00], pseudorandom generators [BFKL93], public key encryption [ABW10], and indistinguishability obfuscation [Lin17], as well as hardness results for learning [DS16] and optimization [Fei02]. Random CSPs also provide a rich testbed for algorithmic and lower-bound techniques based on statistical physics [MM09] and convex relaxation hierarchies [KMOW17, RRS17].

For a random, say, Max-Cut instance average degree d, its optimum value is with high probability (whp) concentrated around a certain function of d. Similarly, given a random 3SAT instance where each variable participates in an average of d clauses, the satisfiability status is whp determined by d. However explicitly working out the optimum/satisfiability as a function of d is usually enormously difficult; see, for example, Ding–Sly–Sun's landmark verification [DSS15] of the kSAT threshold for sufficiently large k, or Talagrand's proof [Tal06] of the Parisi formula for the Sherrington–Kirkpatrick model (Max-Cut with random Gaussian edge weights). The lat-

ter was consequently used by Dembo–Montanari–Sen [DMS17] (see also [Sen18]) to determine that the Max-Cut value in a random d-regular graph is a $\frac{1}{2} + \frac{P^*}{\sqrt{d}}(1 \pm o_d(1))$ fraction of edges (whp), where $P^* \approx .7632$ is an analytic constant arising from Parisi's formula.

Computational gaps for certification. Turning to computational issues, there are two main algorithmic tasks associated with an n-variable CSP: *searching* for an assignment achieving large value (hopefully near to the optimum), and *certifying* (as, e.g., convex relaxations do) that no assignment achieves some larger value. Let's take again the example of random d-regular Max-Cut, where whp we have $\mathrm{OPT} \approx \frac{1}{2} + \frac{P^*}{\sqrt{d}}$. It follows from [Lyo17] there is an efficient algorithm that whp finds a cut of value at least $\frac{1}{2} + \frac{2/\pi}{\sqrt{d}}$. One might say that this provides a $\frac{2}{\pi P^*}$-approximation for the search problem,[1] where $\frac{2}{\pi P^*} \approx .83$. On the other side, the Max-Cut in a d-regular graph G is always at most $\frac{1}{2} + \frac{-\lambda_{\min}(G)}{2d}$, and Friedman's proof of Alon's Conjecture [Fri08] shows that $-\lambda_{\min}(G) \leqslant 2\sqrt{d-1} + o_n(1)$ whp; thus computing the smallest eigenvalue efficiently certifies $\mathrm{OPT} \lessapprox \frac{1}{2} + \frac{1}{\sqrt{d}}$. One might say that this efficient spectral algorithm provides a $\frac{1}{P^*}$-approximation for the certification problem, where $\frac{1}{P^*} \approx 1.31$.

It is a very interesting question whether either of these approximation algorithms can be improved. On one hand, it would seem desirable to have efficient algorithms that come arbitrarily close to matching the "true" answer on random inputs. On the other hand, the nonexistence of such algorithms would be useful for cryptography and hardness-of-approximation and -learning results.

Speaking broadly, efficient algorithms for the search problem seem to do better than efficient algorithms for the certification problem. For example, given a random 3SAT instance with clause density slightly below the satisfiability threshold of ≈ 4.2667, there are algorithms [MPR16] that seem to efficiently find satisfying assignments whp. On the other hand, the longstanding Feige Hypothesis [Fei02] is that efficient algorithms cannot certify unsatisfiability at any large constant clause density, and indeed there is no efficient algorithm that is known to work at density $o(\sqrt{n})$. Similarly, for the Sherrington–Kirkpatrick model, Montanari [Mon19] has recently given an efficient PTAS for the search problem[2], whereas the best known efficient algorithm for the certification problem is again only a $1/P^*$-approximation. These kinds of gaps seem to be closely related to "information-computation gaps" and Kesten–Stigum thresholds for information recovery and planted-CSP problems.

In this work we focus on potential computational thresholds for random CSP certification/refutation problems in the sparse setting, and in particular how these thresholds depend on the "type" of the CSP. For CSPs with a predicate supporting a pairwise-uniform distribution — such as kSAT or kXOR, $k \geqslant 3$ — there is solid evidence that the computational threshold for efficient certification of unsatisfiability is very far from the actual unsatisfiability threshold. Such CSPs are whp unsatisfiable at constant constraint density, but any polynomial-time algorithm using the powerful Sum-of-Squares (SoS) algorithm fails to refute unless the density is $\Omega(\sqrt{n/\log n})$ [KMOW17]. But outside the pairwise-supporting case, and especially for "2XOR-like" CSPs such as Max-Cut

[1] Depending on one's taste in normalization; i.e., whether one prefers the objective function $\mathrm{avg}_{(u,v)\in E}(\frac{1}{2} - \frac{1}{2}x_u x_v)$ or $-\mathrm{avg}_{(u,v)\in E} x_u x_v$, for $x \in \{\pm 1\}^V$.

[2] Modulo a widely believed analytic assumption.

and NAE-3SAT (Not-All-Equal 3SAT), the situation is much more subtle. For one, the potential gaps are much more narrow; e.g., in random NAE-3SAT, even a simple spectral algorithm efficiently refutes satisfiability at constant constraint density. Thus one must look into the actual *constants* to determine if there may be an "information-computation" gap. Another concern is that evidence for computational hardness in the form of SoS lower bounds (degree 4 or higher) seems very hard to come by (see, e.g., [Mon17]).

Prior work. Let us describe two prior efforts towards computational thresholds for upper-bound-certification in "2XOR-like" random CSPs. Montanari and Sen [MS16] (see also [BKM17]) investigated the Max-Cut problem in random d-regular graphs, where the optimum value is $\frac{1}{2} + \frac{P^*}{\sqrt{d}}$ whp (ignoring $1 \pm o_d(1)$ factors). Friedman's Theorem implies that the basic eigenvalue bound efficiently certifies the value is at most $\frac{1}{2} + \frac{1}{\sqrt{d}}$. By using a variant of the Gaussian Wave [Elo09, CGHV15, HV15] construction for the infinite d-ary tree, Montanari and Sen were able to show that even the Goemans–Williamson semidefinite programming (SDP) relaxation [DP93, GW95] is still just $\frac{1}{2} + \frac{1}{\sqrt{d}}$ whp. This may be considered evidence that *no* polynomial-time algorithm can certify upper bounds better than $\frac{1}{2} + \frac{1}{\sqrt{d}}$, as Goemans–Williamson has seemed to be the optimal polynomial-time Max-Cut algorithm in all previous circumstances. Of course it would be more satisfactory to see higher-degree SoS lower bounds, but as mentioned these seem very difficult to come by.

Recently, Deshpande et al. [DMO$^+$19b] have given similar results for random "c-constraint-regular" NAE-3SAT CSPs; i.e., random instances where each variable participates in exactly c NAE-3SAT constraints.[3] Random c-constraint-regular instances of NAE-3SAT are easily shown to be unsatisfiable (whp) for $c \geqslant 8$. Deshpande et al. identified an exact threshold result for when the natural SDP algorithm is able to certify unsatisfiability: it succeeds (whp) if $c > 13.5$ and fails (whp) if $c < 13.5$. Indeed, they show that for $c \geqslant 14$ even the basic spectral algorithm certifies unsatisfiability, whereas for $c \leqslant 13$ even the SDP augmented with "triangle inequalities" fails to certify unsatisfiability. Again, this gives evidence for a gap between the threshold for unsatisfiability and the threshold for computationally efficient refutation. The techniques used by Deshpande et al. are similar to those of Montanari–Sen, except with random (b, c)-biregular graphs replacing random c-regular graphs. (The reason is that the primal graph of a random c-constraint-regular NAE-3SAT instance resembles the square of a random $(3, c)$-biregular graph.)

In fact, the Deshpande et al. result is more refined, being concerned not just with satisfiability of random NAE-3SAT instances, but their optimal value as maximization problems. Letting $f(c) = \frac{9}{8} - \frac{3}{8} \cdot \frac{(\sqrt{c-1}-\sqrt{2})^2}{c}$ for $c \geqslant 3$, they determined that in a random c-constraint-regular NAE-3SAT instance, the SDP value is whp $f(c) \pm o(1)$; and furthermore, this is also the basic eigenvalue bound and the SDP-with-triangle-inequalities bound. (Note that $f(13.5) = 1$.) Again, this may suggest that in these instances, computationally efficient algorithms can only certify that at most an $f(c) + o(1)$ fraction of constraints are simultaneously satisfiable.

[3] We have changed terminology to avoid a potential future confusion; we will be associating NAE-3SAT constraints with triangle graphs, so c-constraint-regular NAE-3SAT instances will be associated to $2c$-regular graphs.

5.1.1 Our results

The goal of the present work is to generalize the preceding Montanari–Sen and Deshpande et al. results to a broader class of sparse random 2CSPs and 2XOR-like optimization problems, obtaining precise values for their SDP values. Along the way, we need to come to a deeper understanding of the combinatorial and analytic tools used (nonbacktracking walks, Ihara–Bass formulas, eigenvalues of random graphs and infinite graphs) and we need to extend these tools to graphs that do *not* locally resemble trees (as in Montanari–Sen and Deshpande et al.). We view this aspect of our work as a main contribution, beyond the mere statement of SDP values for specific CSPs. We defer to Section 5.1.2 more detailed discussions of the technical conditions under which we can obtain Ihara–Bass and Friedman-, and Gaussian Wave-type theorems. But roughly speaking, we are able to analyze the SDP value for random regular instances of optimization problems where each "constraint" (not necessarily a predicate) is an *edge-signed graph with two eigenvalues*. Such constraints include: a single edge (corresponding to random regular Max-Cut or 2XOR as in Montanari–Sen); a complete graph (studied by Deshpande et al., with the K_3 case corresponding to random regular NAE-3SAT); the Sort_4 (a.k.a. CHSH) predicate; and, $\mathsf{Forrelation}_k$ constraints. These last two have motivation from quantum mechanics, and in fact the SDP value of the associated CSPs is precisely their "quantum value". We discuss quantum connections further in Section 5.2.2.

We state here two theorems that our new techniques allow us to prove. Recall the Sort_4 predicate, which is satisfied iff its 4 Boolean inputs x_1, x_2, x_3, x_4 satisfy $x_1 \leqslant x_2 \leqslant x_3 \leqslant x_4$ or $x_1 \geqslant x_2 \geqslant x_3 \geqslant x_4$. We precisely define "random c-constraint-regular CSP instance" in Section 5.2, but in brief, we work in the "random lift" model, each variable participates in exactly c constraints, and each constraint is given random negations.[4]

Theorem 5.1.1. *For random c-constraint-regular instances of the Sort_4-CSP, the SDP-satisfiability threshold occurs (in a sense) at $c = 4 + 2\sqrt{2} \approx 6.83$. Indeed, if $c \geqslant 7$ then even the basic eigenvalue bound certifies unsatisfiability (whp); and, if $c \leqslant 6$ then the basic SDP relaxation fails to certify unsatisfiability (whp).*

We remark that the trivial first-moment calculation shows that a random c-constraint-regular Sort_4-CSP is already unsatisfiable whp at degree $c = 4$. Thus we again have evidence for a gap between the true threshold for unsatisfiability and the efficiently-certifiable threshold.

Generalizing this, the $\mathsf{Forrelation}_k$ constraint is a certain (quantum-inspired) map $\{\pm 1\}^{2^k + 2^k} \to [-1, +1]$ that measures how correlated one k-bit Boolean function is with the Fourier transform of a second k-bit Boolean function. We give precise details in Section 5.2.2; here we just additionally remark that $\mathsf{Forrelation}_1$ corresponds to the "CHSH game", and that $\frac{1}{2} + \mathsf{Forrelation}_1$ is equivalent to the Sort_4 predicate.

Theorem 5.1.2. *For random c-constraint-regular instances of the $\mathsf{Forrelation}_k$-CSP and any constant $\varepsilon > 0$, the SDP value is whp in the range $\frac{2\sqrt{c-1}}{c \cdot 2^{k/2}} \pm \varepsilon$. This is also true of the eigenvalue bound.*

When considering the SDP value for $\frac{1}{2} + \mathsf{Forrelation}_1$, the formula above crosses the threshold of 1 when $c = 4 + 2\sqrt{2}$, yielding the statement in Theorem 5.1.1 about the SDP-satisfiability

[4] Our result holds for either of the following two negation models: (i) each *constraint* is randomly negated; or, (ii) the constraints are not negated, but each constraint is applied to random *literals* rather than random variables.

threshold of random c-constraint-regular Sort_4-CSPs.

5.1.2 Sketch of our techniques

Here we sketch how our results like Theorem 5.1.1 and Theorem 5.1.2 are proven, using random Sort_4-CSPs as a running example. A key property of the Sort_4 predicate is that it is essentially equivalent to the following "2XOR" instance:

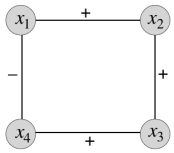

Figure 5.1: The Sort_4 predicate

More precisely, suppose $(x_1, x_2, x_3, x_4) \in \{\pm 1\}^4$ satisfies the Sort_4 predicate. Then in the graph above, exactly 3 out of 4 edges will be "satisfied" — where an edge is considered satisfied when the product of its endpoint-labels equals the edge's label. Conversely, if (x_1, x_2, x_3, x_4) doesn't satisfy Sort_4 then exactly 1 out of the 4 edges above will be satisfied. Now suppose we choose a random n-vertex c-constraint-regular instance $\mathcal{I}$ of the Sort_4-CSP with, say, $c = 2$. A small piece of such an instance might look like the following:[5]

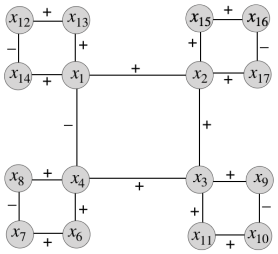

Figure 5.2: Piece of Sort_4 instance

Up to a trivial affine shift in the objective function, the optimization task is now to label the variables/vertices of $\mathcal{I}$ with ± 1 values $x_1, \ldots, x_n$ so as to maximize $\frac{1}{n} \sum_{ij} A_{ij} x_i x_j$, where $A \in \{0, \pm 1\}^{n \times n}$ is the adjacency matrix of the edge-signed graph partially depicted above. The "eigenvalue upper bound" $\mathrm{EIG}(\mathcal{I})$ arises from allowing the x_i's to be arbitrary real numbers, subject to the constraint $\sum_i x_i^2 = n$. The "SDP upper bound" $\mathrm{SDP}(\mathcal{I})$ (which is at least as tight: $\mathrm{SDP}(\mathcal{I}) \leqslant \mathrm{EIG}(\mathcal{I})$) arises from allowing the x_i's to be arbitrary unit vectors in $\mathbb{R}^n$, with the inner product $\langle x_i, x_j \rangle$ replacing $x_i x_j$ in the objective function. Our goal is to identify some quantity $f(c)$ (it will be $\frac{1+\sqrt{2}}{2}$ in the $c = 2$ case) such that

$$\mathrm{EIG}(\mathcal{I}) \lesssim f(c) \lesssim \mathrm{SDP}(\mathcal{I}) \tag{5.1}$$

[5]In fact, since we will have random negations in our instances, some 4-cycles will have three edges labeled -1 and one labeled $+1$, as opposed to the other way around. This is not an important issue for this proof sketch.

up to $1 \pm o(1)$ factors, with high probability. This establishes that all three quantities are equal (up to $1 \pm o(1)$, whp), since $\mathrm{SDP}(\mathcal{I}) \leqslant \mathrm{EIG}(\mathcal{I})$ always.

In this section we mainly describe how to obtain the optimal inequality on the left in (5.1); i.e., how to give a tight bound on the eigenvalues of (the edge-signed graph induced by) $\mathcal{I}$. Notice that if we were studying just random Max-Cut or 2XOR CSPs, we would have to get tight bounds on the eigenvalues of a standard random c-regular graph.[6] Excluding the top eigenvalue of c in the case of Max-Cut, these eigenvalues are (whp) all at most $2\sqrt{c-1} + o_n(1)$ in magnitude. This is thanks to Friedman's (difficult) proof of Alon's Conjecture [Fri08], made moderately less difficult by Bordenave [Bor19]. The "magic number" $2\sqrt{c-1}$ is precisely the spectral radius of the *infinite* c-regular tree — i.e., the infinite graph that random c-regular graphs "locally resemble".

Returning to random 2-constraint-regular instances of the Sort_4-CSP, the (edge-signed) infinite graph X that *they* "locally resemble" is the following:

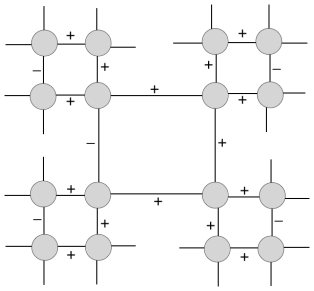

Figure 5.3: Sort_4 infinite graph

Here $X \coloneqq \mathsf{Sort}_4 \dotplus \mathsf{Sort}_4$ is the so-called *additive product* of 2 copies of the Sort_4 graph, a notion introduced in [MO18], which we will formally define in Section 5.2.4. By analogy with Alon's Conjecture, it's natural to guess that the spectral radius of a random 2-constraint-regular Sort_4-CSP instance is whp $\rho(X) \pm o_n(1)$, where $\rho(X)$ denotes the spectral radius of X (which can be shown to be $2\sqrt{2}$). Indeed, our main effort is to prove the upper bound of $\rho(X) + o_n(1)$, thereby establishing the left inequality in (5.1) with $f(c) = \rho(X)$. (As for the right inequality, it can proven using the "Gaussian Wave" idea, allowing one to convert approximate eigenvectors of the infinite graph X to matching SDP solutions on random finite graphs $\mathcal{I}$. We carry this out in Section 5.5.)

Friedman/Bordenave Theorems for two-eigenvalue additive lifts

As stated, our main task in the context of large random 2-constraint-regular Sort_4-CSP instances is to show that their spectral radius is at most $\rho(X) + o_n(1)$ whp. Incidentally, the lower bound of

[6] More precisely, for random Max-Cut we have to lower-bound the smallest eigenvalue; for random 2XOR — which includes randomly negating edges — we have to upper-bound the largest eigenvalue. In the Max-Cut version with no negations, there is the usual annoyance that there is always a first "trivial" eigenvalue of c, and one essentially wants to bound the second-largest (in magnitude) eigenvalue. The effect of random negations is generally to eliminate the trivial eigenvalue, allowing one to focus simply on the spectral radius of the adjacency matrix. This technical convenience is one reason we will always work in a model that includes random negations.

$\rho(X) - o_n(1)$ indeed holds; it follows from a generalization of the "Alon–Boppana Bound" due to Grigorchuk and Żuk [GZ99]. As for the upper bound, the recent work [MO18] implies the analogous "Ramanujan graph" statement; namely, that there *exist* arbitrarily large 2-constraint-regular Sort_4-CSP instances with largest eigenvalue exactly upper-bounded by $\rho(X)$. However we need the analogue of Friedman/Bordenave's Theorem. Unlike in [MO18] we are not able to prove it for arbitrary additive products; we are able to prove it for additive products of "two-eigenvalue" edge-signed graphs. To explain why, we first have to review the proofs of the Alon Conjecture (that c-regular random graphs have their nontrivial eigenvalues bounded by $2\sqrt{c-1} + o_n(1)$).

Both Friedman's and Bordenave's proof of the Alon Conjecture rely on very sophisticated uses of the Trace Method. Roughly speaking, this means counting closed walks of a fixed length k in random c-regular graphs, and (implicitly) comparing these counts to those in the c-regular infinite tree. Actually, both works instead count only *nonbacktracking* walks. The fact that one can relate nonbacktracking walk counts to general walk counts is thanks to an algebraic tool called the *Ihara–Bass Formula* (more on which later); this idea was made more explicit in Bordenave's proof. Incidentally, use of the nonbacktracking walk operator has played a major role in recent algorithmic breakthroughs on community detection and related results (e.g., [KMM$^+$13, MNS18, Mas14, BLM15]).

A reason for passing to nonbacktracking closed walks is that it greatly simplifies the counting. Actually, in the case of the infinite c-regular tree, it *over*simplifies the counting; infinite trees have no nonbacktracking closed walks at all! However, the correct quantity to look at is "almost" nonbacktracking walks of length k, meaning ones that are nonbacktracking for the first $k/2$ steps, and for the last $k/2$ steps, but which may backtrack once right in the middle. There are essentially $(c-1)^{k/2}$ of these in the c-regular infinite tree (one may take $k/2$ arbitrary steps out, but then one must directly walk back home), yielding a value of $((c-1)^{k/2})^{1/k} = \sqrt{c-1}$ for the spectral radius of the nonbacktracking operator of the c-regular infinite tree. Bordenave uses (a very tricky version of) the Trace Method to analogously show that the spectral radius of the nonbacktracking operator of a random c-regular graph is $\sqrt{c-1} + o_n(1)$ whp. Thanks to the Ihara–Bass Formula, this translates into a bound of $2\sqrt{c-1} + o_n(1)$ for the spectral radius of the usual adjacency operator.

Returning now to our scenario of random 2-constraint-regular Sort_4-CSP instances (with their analogous infinite edge-signed graph X), we encounter a severe difficulty. Namely, passing to nonbacktracking walks no longer creates a drastic simplification in the counting, since there are nonbacktracking cycles within the constraint graphs themselves (in our example, 4-cycles graphs).[7] Thus nonbacktracking closed walks in large random instances can have complicated structures, with many internal nonbacktracking cycles.

A saving grace in the case of Sort_4-CSPs, and also ones based on $\mathsf{Forrelation}_k$ or complete-graph constraints for example, is that the adjacency matrices of these graphs have only *two distinct eigenvalues*. (We will also use that their edge weights are ± 1.) For example, after

[7] In fact, since we have edge weights (signs), we need to look at the *weight* (not number) of walks, but the point still stands.

rearranging the variables in the Sort_4 predicate, its adjacency matrix is

$$A = \begin{pmatrix} 0 & 0 & +1 & +1 \\ 0 & 0 & +1 & -1 \\ +1 & +1 & 0 & 0 \\ +1 & -1 & 0 & 0 \end{pmatrix}, \tag{5.2}$$

which has eigenvalues of $\pm\sqrt{2}$ (with multiplicity 2 each). The two-eigenvalue property implies that A satisfies a quadratic equation, and hence any polynomial in A is equivalent to a polynomial of degree *at most* 1. The upshot is that we can relate general walks in Sort_4-CSPs (or more generally, CSPs with two-eigenvalue constraints) to what we call *nomadic* walks: ones that take *at most* 1 consecutive step within a single constraint. Let us make an informal definition (see Section 5.2.4 for a formal definition):

Definition 5.1.3. Given a finite CSP graph, the *nomadic walk operator* B is a matrix indexed by the directed edges in the graph. Its $B[e, e']$ entry is equal to the edge-weight of e' provided:

- (e, e') forms an oriented length-2 path; and,
- e and e' come from *different* constraints.

Otherwise the $B[e, e']$ entry is 0. This operator generalizes the nonbacktracking walk operator for Max-Cut/2XOR graphs in which each undirected edge is considered to be a single "constraint".

The utility of this nomadic walk operator is twofold for us. First, for two-eigenvalue CSPs we can relate the eigenvalues of the usual adjacency operator to those of the nomadic walk operator through the following generalization of the Ihara–Bass Formula:

Theorem 5.1.4 (informal)**.** *Let A be the adjacency matrix and B the nomadic walk operator of a finite c-constraint-regular CSP graph on n vertices, where each predicate has exactly 2 distinct eigenvalues: λ_1 and λ_2. Define $L(t) := \mathbb{1} - At + (\lambda_1 + \lambda_2)t\mathbb{1} + (c-1)(-\lambda_1\lambda_2)t^2$. Then we have*

$$(1+\lambda_1 t)^{n\frac{c\lambda_2}{\lambda_2-\lambda_1}-1}(1+\lambda_2 t)^{n\frac{c\lambda_1}{\lambda_1-\lambda_2}-1}\det L(t) = \det(\mathbb{1} - Bt).$$

We prove Theorem 5.1.4 in Section 5.3. In the remaining discussion below, we let B be the nomadic walk operator of a random c-constraint-regular CSP graph on n vertices, where the precise random model is given in Definition 5.2.18. Further, we assume that the predicate of the CSP has two distinct eigenvalues: λ_1 and λ_2.

The second utility of nomadic walks is that they provide the key simplification needed to make closed-walk counting in non-tree-like CSPs tractable. Because of this, we are able to establish the following modification of Bordenave's proof of Friedman's Theorem in Section 5.6:

Theorem 5.1.5. *With high probability,*

$$\rho(B) \leqslant \sqrt{(c-1)(-\lambda_1\lambda_2)} + o_n(1).$$

And we can use our version of Ihara–Bass, Theorem 5.1.4, to conclude bounds on the spectrum of the adjacency matrix A from Theorem 5.1.5, which is worked out in Section 5.4.

Theorem 5.1.6. *With high probability,*

$$\operatorname{Spec}(A) \subseteq \left[\lambda_1 + \lambda_2 - 2\sqrt{(c-1)(-\lambda_1\lambda_2)} - o(1), \lambda_1 + \lambda_2 + 2\sqrt{(c-1)(-\lambda_1\lambda_2)} + o(1)\right].$$

Yet another advantage of using nomadic walks instead of closed walks is that in Theorem 5.1.6 we are able to bound the left and right spectral edge of A by *different* values, whereas counting closed walks would, at best, only give an upper bound on $|\lambda|_{\max}(A)$.

Theorem 5.1.6 lets us conclude an upper bound on the SDP value, and we complement that with a lower bound via the construction of an SDP solution that nearly matches the upper bound. In particular, we prove the following in Section 5.5.

Theorem 5.1.7. *For every $\varepsilon > 0$, whp there exists a PSD matrix M with an all-ones diagonal such that*

$$\langle A, M\rangle \geqslant \left(\lambda_1 + \lambda_2 + 2\sqrt{(c-1)(-\lambda_1\lambda_2)} - \varepsilon\right) n.$$

As detailed out in Section 5.7, this lets us conclude the main theorem of this chapter:

Theorem 5.1.8. *For random c-constraint-regular instances of a CSP with 2 distinct eigenvalues λ_1 and λ_2, the SDP value is in the range*

$$\frac{\lambda_1 + \lambda_2 + 2\sqrt{(c-1)(-\lambda_1\lambda_2)}}{c(-\lambda_1\lambda_2)} \pm \varepsilon$$

with high probability, for any $\varepsilon > 0$.

Theorem 5.1.2 can be viewed as a special case of Theorem 5.1.8.

5.2 Preliminaries

5.2.1 2XOR optimization problems and their relaxations

All of the CSPs studied in this work (Max-Cut, NAE-3SAT, Sort_4, $\mathsf{Forrelation}_k$, etc.) will effectively reduce to 2XOR *optimization problems* — equivalently, the problem maximizing a homogeneous degree-2 polynomial with ± 1 coefficients over the Boolean hypercube.

Definition 5.2.1. (Optimization of 2XOR instances) Let $G = (V, E)$ be an undirected graph (possibly with parallel edges), with edge-signing $\mathrm{wt} : E \to \{\pm 1\}$. We call the pair $\mathcal{I} = (G, \mathrm{wt})$ an *instance*. The associated 2XOR *optimization problem* is to determine the *(true) optimum value*

$$\mathrm{OPT}(\mathcal{I}) = \max_{x:V\to\{\pm1\}} \operatorname*{avg}_{e=\{u,v\}\in E} \{\mathrm{wt}(e)x_u x_v\} \in [-1, +1].$$

The special case in which $\mathrm{wt} \equiv -1$ is referred to as the Max-Cut problem on G, as in this case $\frac{1}{2} + \frac{1}{2}\mathrm{OPT}(\mathcal{I}) = \mathsf{Max\text{-}Cut}(G)$, the maximum fraction of edges that can be cut by a bipartition of V.

Determining $\mathrm{OPT}(\mathcal{I})$ is NP-hard in the worst case, leading to the study of computationally tractable approximations/relaxations. Two such approximations are the *eigenvalue bound* and the *SDP bound*, which we now recall.

Definition 5.2.2. (Adjacency matrix/operator) The *adjacency matrix* A of a finite weighted graph (G, wt) has rows and columns indexed by V; the entry $A[u, v]$ equals the sum of $\mathrm{wt}(e)$ over all edges with endpoints $\{u, v\}$. In case G is infinite we can more generally define the adjacency operator A on $\ell_2(V)$ as follows:

$$\text{for } F \in \ell_2(V), \quad AF(u) = \sum_{e=(u,v)\in E} \mathrm{wt}(e)F(v).$$

Definition 5.2.3. (Eigenvalue bound) The *eigenvalue bound* $\mathrm{EIG}(\mathcal{I})$ for 2XOR instance $\mathcal{I}$ with adjacency matrix A is $\frac{n}{2|E|}\lambda_{\max}(A)$, where $\lambda_{\max}$ denotes the maximum eigenvalue. We have $\mathrm{OPT}(\mathcal{I}) \leqslant \mathrm{EIG}(\mathcal{I})$ always, as the eigenvalue bound captures the relaxation of 2XOR optimization where we allow any $x : V \to \mathbb{R}$ satisfying $\|x\|^2 = n$.

The *SDP value* provides an even tighter upper bound on $\mathrm{OPT}(\mathcal{I})$, and is still efficiently computable.[8] The SDP bound dates back to Lovász's Theta Function in the context of the IndependentSet problem [Lov79], and was proposed in the context of the Max-Cut problem by Delorme and Poljak [DP93].

Definition 5.2.4. (SDP bound) The *SDP bound* $\mathrm{SDP}(\mathcal{I})$ for 2XOR instance $\mathcal{I}$ is

$$\mathrm{SDP}(\mathcal{I}) = \max_{\vec{x}:V\to S^{m-1}} \operatorname*{avg}_{e=\{u,v\}\in E} \{\mathrm{wt}(e)\langle \vec{x}_u, \vec{x}_v\rangle\} \in [-1,+1],$$

where S^{m-1} refers to the set of unit vectors in $\mathbb{R}^m$ and the maximum is also over m (though $m = n$ is sufficient). The following holds for all $\mathcal{I}$:

$$\mathrm{OPT}(\mathcal{I}) \leqslant \mathrm{SDP}(\mathcal{I}) \leqslant \mathrm{EIG}(\mathcal{I}).$$

The left inequality is obvious. One way to see the right inequality is to use the fact [DP93], based on SDP duality, that $\mathrm{SDP}(\mathcal{I})$ is also equal to the minimum value of the eigenvalue bound applied to $A + Y$, where A is the adjacency matrix and Y ranges over all matrices of trace 0.

Goemans and Williamson [GW95] famously showed that

$$\tfrac{1}{2} + \tfrac{1}{2}\mathrm{SDP}(\mathcal{I}) \leqslant 1.138(\tfrac{1}{2} + \tfrac{1}{2}\mathrm{OPT}(\mathcal{I}))$$

holds for every 2XOR instance, and Feige–Schechtman [FS02] showed their bound can be tight in the worst case.[9] As for directly comparing $\mathrm{SDP}(\mathcal{I})$ and $\mathrm{OPT}(\mathcal{I})$, we have the following:

- ([CW04]) $\mathrm{SDP}(\mathcal{I}) \leqslant O(\mathrm{OPT}(\mathcal{I}) \cdot \log(1/\mathrm{OPT}(\mathcal{I})))$ always holds.
- When G is bipartite (a special case of particular interest, see Section 5.2.2), it holds that $\mathrm{SDP}(\mathcal{I}) \leqslant K \cdot \mathrm{OPT}(\mathcal{I})$ for constant K. This is known as *Grothendieck's inequality* [Gro53], and the constant is known [BMMN13] to satisfy $K < \pi/(2\ln(1+\sqrt{2})) \approx 1.78$.

5.2.2 Quantum games, and some quantum-relevant constraints

In the case when the underlying graph G is bipartite, $\mathrm{SDP}(\mathcal{I})$ has another important interpretation: it is the true *quantum* value of the 2-player 1-round "nonlocal game" associated to $\mathcal{I}$. We give definitions below, but let us mention that the Sort_4 (equivalently, CHSH) and $\mathsf{Forrelation}_k$ constraints from Theorem 5.1.1 and Theorem 5.1.2 are both: (a) bipartite; (b) directly inspired by quantum theory. Thus those two theorems can be interpreted as determining the true quantum value of random c-constraint-regular nonlocal games based on CHSH and $\mathsf{Forrelation}_k$.

Let us now recall the relevant quantum facts.

[8] More precisely, it can be computed to within $\pm\epsilon$ in $\mathrm{poly}(|\mathcal{I}|, \log(1/\varepsilon))$ time using the Ellipsoid Algorithm [GLS88, DP93].

[9] The case of Max-Cut on the 5-cycle — i.e., maximizing $-\frac{1}{5}(x_1x_2 + x_2x_3 + x_3x_4 + x_4x_5 + x_5x_1)$ on $\{\pm1\}^5$ — already has $\mathrm{OPT} = 3/5$ and $\mathrm{SDP} = (1+\sqrt{5})/4$, showing that 1.138 cannot be improved below 1.131.

Definition 5.2.5 (Nonlocal 2XOR games)**.** Given a 2XOR instance $\mathcal{I} = (G, \mathrm{wt})$ with $G = (U, V, E)$ bipartite, the associated *nonlocal (*2XOR*) game* is the following. There are spatially separated players Alice and Bob. A referee chooses $e = (u, v) \in E$ uniformly at random, tells u to Alice, and tells v to Bob. Without communicating, Alice and Bob are required to respond with signs $x_u, y_v \in \{\pm 1\}$. The *value* to the players is the expected value of $\mathrm{wt}(e)x_u y_v$. It is easy to see that if Alice and Bob are deterministic, or are allowed classical shared randomness, then the optimum value they can achieve is precisely $\mathrm{OPT}(\mathcal{I})$.

Theorem 5.2.6. *([CHTW04, Tsi80].) In a nonlocal* 2XOR *game, if Alice and Bob are allowed to share unlimited quantumly entangled particles, then the optimal value they can achieve is precisely* $\mathrm{SDP}(\mathcal{I})$.

The fact that there exist bipartite edge-signed $\mathcal{I}$ for which $\mathrm{SDP}(\mathcal{I}) > \mathrm{OPT}(\mathcal{I})$ is foundational for the experimental verification of quantum mechanics, as the following example attests:

Example 5.2.7. Consider the 2XOR instance depicted in Figure 5.4, called CHSH after Clauser, Horne, Shimony, and Holt [CHSH69]. It has

$$\mathrm{OPT}(\mathsf{CHSH}) = 1/2 < 1/\sqrt{2} = \mathrm{SDP}(\mathsf{CHSH}).$$

The upper bound $4 \cdot \mathrm{OPT}(\mathsf{CHSH}) \leqslant 2$ is often called *Bell's inequality* [Bel64], and the higher

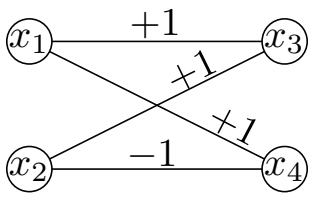

Figure 5.4: The CHSH game/CSP

lower bound $1/\sqrt{2} \leqslant \mathrm{SDP}(\mathsf{CHSH})$ is from [CHSH69] (with $\mathrm{SDP}(\mathsf{CHSH}) \leqslant 1/\sqrt{2}$ due to Tsirelson [Tsi80]). Aspect and others [ADR82] famously experimentally realized this gap between what can be achieved with classical vs. quantum resources.

In fact, the CHSH instance is nothing more than the Sort_4 predicate in disguise! More precisely (cf. (5.2)),

$$\mathsf{CHSH}(x_1, x_2, x_3, x_4) = \tfrac{1}{4}(x_1x_3 + x_2x_3 + x_1x_4 - x_2x_4) = \mathsf{Sort}_4(x_2, x_3, x_1, x_4) - \tfrac{1}{2}.$$

Thanks to its degree-2 Fourier expansion, CSPs based on the Sort_4/CHSH constraint have been studied in a variety of contexts, including concrete complexity [Amb06, APV16, OST+14] and fixed parameter algorithms [Wil07].

Though Sort_4 is a "predicate", in the sense that it takes 0/1 (unsat/sat) values, there's nothing necessary about basing a large CSP on predicates. An interesting family of constraints that can be modeled by 2XOR optimization, originally arising in quantum complexity theory [AA15], is the family of "Forrelation" functions. For any $k \in \mathbb{N}$, the $\mathsf{Forrelation}_k$ function is defined by

$$\mathsf{Forrelation}_k : \{\pm 1\}^{2^k} \times \{\pm 1\}^{2^k} \to [-1, +1], \qquad \mathsf{Forrelation}_k(x_1, \dots, x_{2^k}, y_1, \dots, y_{2^k}) = 2^{-2k} x^\top H_k y,$$

where $H_k = \begin{pmatrix} +1 & +1 \\ +1 & -1 \end{pmatrix}^{\otimes k}$ is the kth Walsh–Hadamard matrix. Note that $\mathsf{Forrelation}_0$ corresponds to the single-(positive-)edge 2XOR CSP, and $\mathsf{Forrelation}_1$ is CHSH.

5.2.3 2XOR graphs with only 2 distinct eigenvalues

As mentioned, the class of constraints that we treat in this work are those that can be modeled as 2XOR instances with 2 *distinct eigenvalues*. The $\mathsf{Forrelation}_k$ constraint is a prime example; when viewed as an edge-signed graph (i.e., ignoring the 2^{-2k} scaling factors), its eigenvalues are all $\pm 2^{k/2}$. Another example is the complete graph constraint on r variables, which has eigenvalues of $r-1$ and -1 (the latter with multiplicity $r-1$). The $r=3$ complete-graph case, after a trivial affine shift, also corresponds to a Boolean predicate that is well known in the context of CSPs: the NAE-3SAT predicate, as studied in [DMO+19b]. This is because

$$\mathsf{NAE\text{-}3SAT}(x_1, x_2, x_3) = \frac{3}{4} - \frac{3}{4}(x_1x_2 + x_2x_3 + x_3x_1).$$

Let us make some definitions we will use throughout this chapter.

Definition 5.2.8 (2-eigenvalue graphs)**.** We call an undirected, edge-weighted simple graph $\mathcal{I}$ a *2-eigenvalue graph* if there are two real numbers λ_1 and λ_2 such that each eigenvalue of $\mathcal{I}$'s (signed) adjacency matrix A is equal to either λ_1 or λ_2.

See, e.g., [Ram15] for a paper studying such graphs. In this section, let us use the notation from Definition 5.2.8 and prove some properties that will be used throughout this chapter.

First, since A is symmetric, its eigenvectors are spanning and therefore every vector can be written as the sum of a vector in $\ker(A - \lambda_1 \mathbb{1})$ and one in $\ker(A - \lambda_2 \mathbb{1})$. Thus:

Proposition 5.2.9. $(A - \lambda_1\mathbb{1})(A - \lambda_2\mathbb{1}) = 0$*, where* $\mathbb{1}$ *denotes the identity matrix.*

This proposition implies that $A^2 = (\lambda_1 + \lambda_2)A - \lambda_1\lambda_2\mathbb{1}$. Thus we can deduce the following two facts:

Fact 5.2.10. *For any* $v \in V(G)$, $\sum_{u \in V(G)} A[u,v]^2 = A^2[v,v] = -\lambda_1\lambda_2$.

Fact 5.2.11. *For any pair of distinct vertices* $u, v \in V(G)$,

$$\sum_{w \in V(G)} A[u,w]A[w,v] = A^2[u,v] = (\lambda_1 + \lambda_2)A[u,v].$$

5.2.4 Random constraint graphs, instance graphs, and additive products

Definition 5.2.12 (Constraint graphs)**.** An *r-ary, c-atom constraint graph* is any n-fold lift $\mathcal{H}$ of the complete bipartite graph $K_{r,c}$. Each vertex on the c-regular side is called a *variable vertex*, and is typically depicted by a circle. The variable vertices are partitioned into r *variable groups* each of size n, called the 1*st variable group*, the 2*nd variable group*, etc. Each vertex on the r-regular side is called a *constraint* (or *atom*) *vertex*, and is typically depicted by a square. Again, the constraint vertices are partitioned into c *constraint* (or *atom*) groups of size n, called the 1*st constraint/atom group*, 2*nd constraint/atom group*, etc. When $n = 1$, we call $\mathcal{H}$ a *base constraint graph*. We also allow "$n = \infty$": this means we take the infinite (r,c)-biregular tree and partition

its variable vertices into r groups and its constraint variables into c groups in such a way that every variable vertex in the ith group has exactly one neighbor from each of the c constraint groups, and similarly every constraint vertex in the jth group has exactly one neighbor from each of the r variable groups. An example of a constraint graph is shown in Figure 5.6. [10]

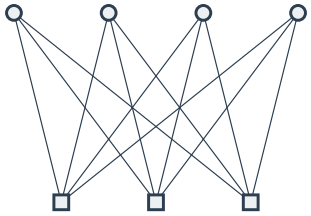

Figure 5.5: The complete $K_{4,3}$ graph

Definition 5.2.13 (Instance graphs)**.** Let $\mathcal{A} = (A_1, \ldots, A_c)$ be a sequence of *atoms*, meaning edge-weighted undirected graphs on a common vertex set $[r]$. (In this chapter, the edge-weights will usually be ± 1.) We also think of each atom as a collection of "2XOR-constraints" on variable set r. Now given an r-ary, c-atom constraint graph $\mathcal{H}$, we can combine it with the atom specification $\mathcal{A}$ to form the *instance graph* $\mathcal{I} \coloneqq \mathcal{A}(\mathcal{H})$. This edge-weighted undirected graph $\mathcal{I}$ has as its vertex set all the variable vertices of $\mathcal{H}$. The edges of $\mathcal{I}$ are formed as follows: We iterate through each $j \in [c]$ and each constraint vertex f in the jth constraint group of $\mathcal{H}$. Given f, with variables neighbors $v_1, \ldots, v_r$ in $\mathcal{H}$, we place a copy of atom A_j onto these vertices in $\mathcal{I}$. ($\mathcal{I}$ may end up with parallel edges.) We refer to the graph obtained by placing a copy of A_j on vertices $v_1, \ldots, v_r$ as A_f, and for any edge e in $\mathcal{I}$ that came from placing A_j, we define $\mathrm{Atom}(e) \coloneqq A_f$. We use $v \sim A_f$ to denote that v is one of $v_1, \ldots, v_r$. For $u, v \in \{v_1, \ldots, v_r\}$, $A_f(u, v)$ denotes the edge in A_f between u and v. And finally, denote the set $\{A_f : f \text{ constraint vertex in } \mathcal{H}\}$ with $\mathrm{Atoms}(\mathcal{I})$. An example of an instance graph and corresponding constraint graph is shown in Figure 5.6.

Remark 5.2.14. Forming $\mathcal{I}$ from $\mathcal{H}$ is somewhat similar to squaring $\mathcal{H}$ (in the graph-theoretic sense) and then restricting to the variable vertices. With this in mind, here is an alternate way to describe the edges of $\mathcal{I}$: For each pair of distinct vertices v, v' in $\mathcal{I}$ (in variable groups i and i', respectively) we consider all length-2 paths joining v and v' in $\mathcal{H}$. For each such path passing through a constraint vertex in constraint group j, we add the edge (v, v') into $\mathcal{I}$ with edge-weight $A_j[i, i']$ (which may be 0).

Remark 5.2.15. We treat atoms as edge-weighted, undirected, complete graphs. Thus, for a constraint vertex f in constraint-graph $\mathcal{H}$, if there is an edge between vertices u and v, and an edge between vertices v and w in the atom A_f, then there is an edge between u and w in A_f. This view is significant in light of the proof of Theorem 5.3.1.

The following notions of additive lifts and additive products were introduced in [MO18]:

Definition 5.2.16 (Random additive lifts)**.** In the context of r-ary, c-atom constraint graphs, a *random n-lifted constraint graph* simply means a usual random n-lift $\mathcal{H}$ (see, e.g., [BL06]) of the

[10] This can be done in an arbitrary "greedy" way, fixing any, say, constraint vertex to be in "group 1", fixing its variables neighbors to be in groups $1 \ldots r$ in an arbitrary way, fixing *their* constraint neighbors to be in groups $2 \ldots c$ in an arbitrary way, etc.

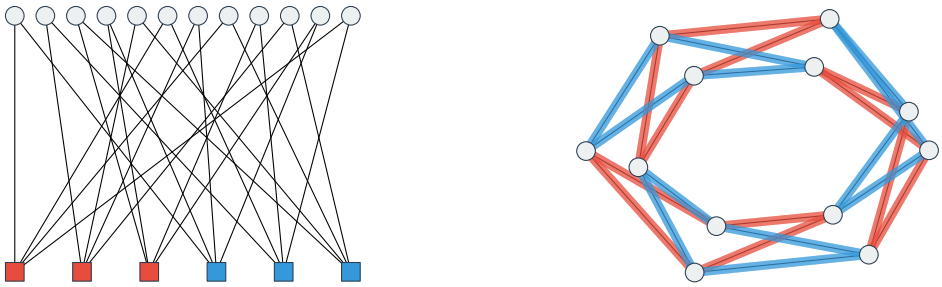

Figure 5.6: The figure on the left shows an example of a 4-ary, 2-atom 3-fold lift constraint graph, with the left bipartition color coded by constraint/atom groups. The figure on the right is the corresponding instance graph on (C_4, C_4), two four-cycle graphs, where each atom is color coded to match the figure on the left.

base constraint graph. Given atoms $\mathcal{A} = (A_1, \ldots, A_c)$, the resulting instance graph $\mathcal{I} = \mathcal{A}(\mathcal{H})$ is called a *random additive lift* of $\mathcal{A}$.

Definition 5.2.17 (Additive products)**.** If instead $\mathcal{H}$ is the "∞-lift" of $K_{r,c}$, the resulting infinite instance graph $\mathcal{I} = \mathcal{A}(\mathcal{H})$ is called the *additive product* of $A_1, \ldots, A_c$, denoted $A_1 \dotplus A_2 \dotplus \cdots \dotplus A_c$.

We will also extend Definition 5.2.13 to allow random additive lifts with *negations*. Eventually we will define a general notion of "1-wise uniform negations", but let us begin with two special cases. In the "constraint negation" model, we assign to each constraint vertex f in $\mathcal{H}$ (from group j) an independent uniformly random sign ξ^f. Then, when the instance graph $\mathcal{I}$ is formed from $\mathcal{H}$, each edge engendered by the constraint f has its weight multiplied by ξ^f. (Thus the edges in this copy of the atom A_j are either all left alone or they are simultaneously negated, with equal probability.) In the "variable negation" model, for each group-j constraint vertex f, adjacent to variable vertices $v_1, \ldots, v_r$, we assign independent and uniformly random signs $(\xi_i^f)_{i \in [r]}$ to the variables. Then when the copy of A_j is added into $\mathcal{I}$, the $\{i, i'\}$-edge has its weight multiplied by $\xi_i^f \xi_{i'}^f$. This corresponds to the constraint being applied to random *literals*, rather than variables.

Notice that in both of these negation models, every time a copy of atom A_j is placed into $\mathcal{I}$, its edges are multiplied by a collection of random signs $(\xi_{ij}^f)_{i,j \in [r]}$ which are "1-wise uniform". This is the only property we will require of a negation model.

Definition 5.2.18 (Random additive lifts with negations)**.** A random additive lift *with* 1*-wise uniform negations* is a variant of Definition 5.2.13 where, for each constraint vertex f there are associated random signs $\xi_i^{(f)} \in \{\pm 1\}$, where $i \in [r]$. For each fixed f, the random variables $\xi_i^{(f)}$ are required to be ± 1 with probability $1/2$ each, but they may be arbitrarily correlated; across different f's, the collections $(\xi_i^{(f)})_{i \in [r]}$ must be independent. When the instance graph $\mathcal{I}$ is formed as $\mathcal{A}(\mathcal{H})$, and a copy of A_j placed into $\mathcal{I}$ thanks to constraint vertex f, each new edge $\{i, i'\}$ has its weight $A_j[i, i']$ multiplied by $\xi_{ii'}^{(f)} := \xi_i^{(f)} \xi_{i'}^{(f)}$.

Remark 5.2.19. For a given constraint-vertex f of an instance graph $\mathcal{I}$ obtained via a random

additive lift with negations, the matrix $\mathrm{Adj}(A_f)$ has the same spectrum as $\mathrm{Adj}(\overline{A_f})$ where $\overline{A_f}$ denotes the subgraph prior to applying random negations, since there is a sign diagonal matrix D such that $\mathrm{Adj}(\overline{A_f}) = D \cdot \mathrm{Adj}(A_f) \cdot D^\dagger$.

5.2.5 Nomadic walks operators

Definition 5.2.20 (Nomadic walks)**.** Let $\mathcal{H}$ be a constraint graph, $\mathcal{A} = (A_1, \ldots, A_c)$ a sequence of atoms, and $\mathcal{I} = \mathcal{A}(\mathcal{H})$ the associated instance graph. For initial simplicity, assume the atoms are unweighted (i.e., all edge weights are $+1$). A *nomadic walk* in $\mathcal{I}$ is a walk where consecutive steps are prohibited from "being in the same atom". Note that if $r = 2$ and the atoms are single edges, a nomadic walk in $\mathcal{I}$ is equivalent to a nonbacktracking walk.

To make the definition completely precise requires "remembering" the constraint graph structure $\mathcal{H}$. Each step along an edge of $\mathcal{I}$ corresponds to taking two consecutive steps in $\mathcal{H}$ (starting and ending at a variable vertex). The walk in $\mathcal{I}$ is said to be nomadic precisely when the associated walk in $\mathcal{H}$ is nonbacktracking.

Finally, in the general case when the atoms A_j have weights, each *walk* in $\mathcal{I}$ gets a weight equal to the product of the edge-weights used along the walk.

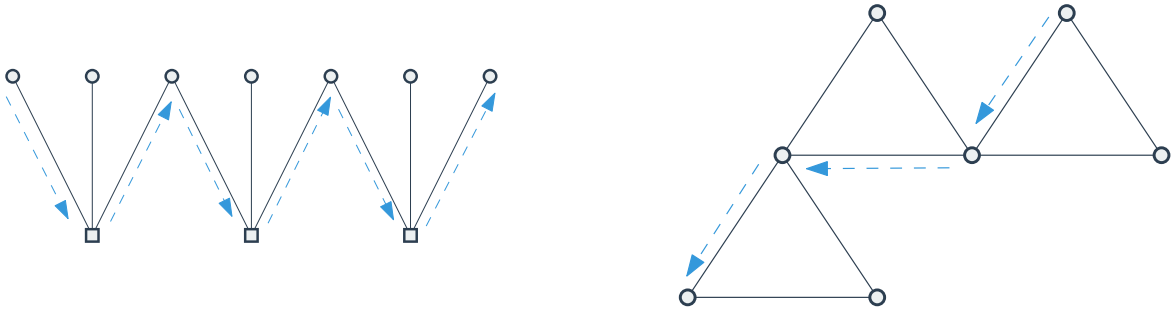

Figure 5.7: The figure on the left shows a nonbacktracking walk on a subset of a 3-ary constraint graph and the one on the right the same nomadic walk on the corresponding instance graph.

Definition 5.2.21 (Nomadic walk operator)**.** In the setting of the previous definition, the *nomadic walk operator* B for $\mathcal{I}$ is defined as follows. Each edge $e = \{u, v\}$ in $\mathcal{I}$ is regarded as two opposing directed edges $\vec{e} = (u, v)$ and $\vec{e}^{-1} = (v, u)$, each having the same edge-weight as e; i.e., $\mathrm{wt}(\vec{e}) = \mathrm{wt}(\vec{e}^{-1}) = \mathrm{wt}(e)$. Let $\vec{E}$ denote the collection of all directed edges. Now B is defined to be the following linear operator on $\ell_2(\vec{E})$:

$$\text{for } F \in \ell_2(\vec{E}), \quad BF(\vec{e}) = \sum_{\vec{e}'} \mathrm{wt}(\vec{e}')F(\vec{e}'),$$

where the sum is over all directed edges $\vec{e}'$ such that the pair $(\vec{e}, \vec{e}')$ forms a nomadic walk of length-2. In the finite-graph case we also think of B as a matrix; the entry $B[\vec{e}, \vec{e}'] = \mathrm{wt}(\vec{e}')$ whenever $(\vec{e}, \vec{e}')$ is a length-2 nomadic walk. Again, in the case where $r = 2$ and all atoms are single edges, the nomadic walk operator B coincides with the nonbacktracking walk operator. (See, e.g., [AFH15] for more on nonbacktracking walks operators.)

5.2.6 Operator theory

The results in this section can be found in a standard textbook on functional analysis or operator theory (see, for e.g. [Kub12]).

Let V be an some countable set and let $T : \ell_2(V) \to \ell_2(V)$ be a bounded, self-adjoint linear operator.

Definition 5.2.22. We refer to the *spectrum* of T, $\mathrm{Spec}(T)$, as the set of all complex λ such that $\lambda\mathbb{1} - T$ is not invertible. $\mathrm{Spec}(T)$ is a nonempty, compact set.

Definition 5.2.23. We call λ an *approximate eigenvalue* of T if for every $\varepsilon > 0$, there is unit x in $\mathcal{X}$ such that $\|Tx - \lambda x\| \leqslant \varepsilon$. We call such an x an *ε-approximate eigenvector* or *ε-approximate eigenfunction*.

Theorem 5.2.24. *If T is a self-adjoint operator, then every $\lambda \in \mathrm{Spec}(T)$ is an approximate eigenvalue.*

Theorem 5.2.25. *[Consequence of Proposition 4.L of [Kub12]] If λ is an isolated point in $\mathrm{Spec}(T)$, then it is an eigenvalue of T, i.e., it is a 0-approximate eigenvalue.*

Corollary 5.2.26. $\lambda_{\min} := \min\{\mathrm{Spec}(T)\}$ *and* $\lambda_{\max} := \max\{\mathrm{Spec}(T)\}$ *are both approximate eigenvalues of T.*

Fact 5.2.27. *Additionally,*

$$\lambda_{\min}(T) = \inf_{\|x\|=1} \langle x, Tx\rangle,$$
$$\lambda_{\max}(T) = \sup_{\|x\|=1} \langle x, Tx\rangle.$$

Definition 5.2.28. The *spectral radius* $\rho(T)$ is defined as $\max_{\sigma\in\mathrm{Spec}(T)} |\sigma|$.

Definition 5.2.29. The *operator norm* of T, denoted $\|T\|_{\mathrm{op}}$, is defined as

$$\sup_{\|x\|=1,\|y\|=1} \langle y, Tx\rangle = \sup_{\|x\|=1} \|Tx\|.$$

Fact 5.2.30. $\rho(T) = \lim_{k\to\infty} \|T^k\|_{\mathrm{op}}^{1/k}$.

5.3 An Ihara–Bass formula for additive lifts of 2-eigenvalue atoms

Let $\mathcal{A}$ be a sequence of atoms such that every atom has the same pair of exactly two distinct eigenvalues, λ_1 and λ_2, and let $\mathcal{H}$ be a constraint graph on variable set V. Let $\mathcal{I} = \mathcal{A}(\mathcal{H})$ be the corresponding instance graph. In this section, we use A and B to refer to the adjacency matrix and nomadic walk matrix respectively of $\mathcal{I}$. The vertex set of $\mathcal{I}$ is V. This section is devoted to proving our generalization of the Ihara–Bass formula, stated below.

Theorem 5.3.1. *Let $L(t) := \mathbb{1} - At + (\lambda_1 + \lambda_2)t\mathbb{1} + (c-1)(-\lambda_1\lambda_2)t^2$. Then we have*

$$(1+\lambda_1 t)^{|V|\frac{c\lambda_2}{\lambda_2-\lambda_1}-1}(1+\lambda_2 t)^{|V|\frac{c\lambda_1}{\lambda_1-\lambda_2}-1} \det L(t) = \det(\mathbb{1} - Bt).$$

Our proof is a modification of one of the proofs of the Ihara–Bass formula from [Nor97].

Nomadic Polynomials. Our first step is to define the following sequence of polynomials.

$$\begin{aligned}
&p_0(x) = 1\\
&p_1(x) = x\\
&p_2(x) = x^2 - (\lambda_1+\lambda_2)x - c(-\lambda_1\lambda_2)\\
&p_k(x) = xp_{k-1}(x) - (\lambda_1+\lambda_2)p_{k-1}(x) - (c-1)(-\lambda_1\lambda_2)p_{k-2}(x) \qquad \text{for } k \geqslant 3
\end{aligned}$$

and introduce the key player in the proof: the matrix of generating functions $F(t)$ defined by

$$F(t)_{u,v} = \sum_{k\geqslant 0} p_k(A)t^k.$$

We use $\mathrm{wt}(e)$ to denote the weight on edge e, and define the weight of a walk $W = e_1e_2\dots e_\ell$ as

$$\mathrm{wt}(W) := \prod_{i=1}^{\ell} \mathrm{wt}(e_i).$$

We first establish combinatorial meaning for the polynomials $p_k(A)$.

Claim 5.3.2. $p_k(A)_{uv}$ is equal to the total weight of nomadic walks of length k from u to v.

Proof. When $k = 0$ and 1, the claim is clear. We proceed by induction.

Supposing the claim is indeed true for $p_s(A)$ when $s \leqslant k-1$, then $Ap_{k-1}(A)_{uv}$ is the total weight of length-k walks from u to v whose first $k-1$ steps are nomadic and whose last step is arbitrary. Call the collection of these walks $\mathcal{W}_{uv}$. For $W \in \mathcal{W}_{uv}$, let W_i denote the edge walked on by the i-th step of W and let $W_{(i)}$ denote the length-i walk obtained by taking the length-i prefix of W. We use lowercase w_i to denote the vertex visited by the ith step of the walk. Each $W \in \mathcal{W}_{uv}$ falls into one of the following three categories.

1. W is a nomadic walk. Call the collection of these walks $\mathcal{W}^{(1)}_{uv}$.
2. $W_k = W_{k-1}^{-1}$. Call the collection of these walks $\mathcal{W}^{(2)}_{uv}$.
3. W_{k-1} and W_k are in the same atom but $W_k \neq W_{k-1}^{-1}$. Call the collection of these walks $\mathcal{W}^{(3)}_{uv}$.

Suppose $k \geqslant 3$.

$$\begin{aligned}
\sum_{W\in\mathcal{W}^{(2)}_{uv}} \mathrm{wt}(W) &= \sum_{W\in\mathcal{W}^{(2)}_{uv}} \mathrm{wt}(W_{k-1})\mathrm{wt}(W_{k-1}^{-1})\mathrm{wt}(W_{(k-2)})\\
&= \sum_{W\in\mathcal{W}^{(2)}_{uv}} \mathrm{wt}(W_{k-1})^2\mathrm{wt}(W_{(k-2)})\\
&= \sum_{\substack{W' \ (k-2)\text{-length nomadic walk}\\ \text{from } u \text{ to } v}} \mathrm{wt}(W') \sum_{e\notin \mathrm{Atom}(W'_{k-2})} \mathrm{wt}(e)^2
\end{aligned}$$

We apply Fact 5.2.10 and get

$$= \sum_{\substack{W' \ (k-2)\text{-length nomadic walk}\\ \text{from } u \text{ to } v}} \mathrm{wt}(W')(c-1)(-\lambda_1\lambda_2)$$

$$= (c-1)(-\lambda_1\lambda_2)p_{k-2}(A)_{uv}.$$

An identical argument shows that when $k = 2$,

$$\sum_{W \in \mathcal{W}_{uv}^{(2)}} \mathrm{wt}(W) = c(-\lambda_1\lambda_2)$$

We do a similar calculation for $\mathcal{W}_{uv}^{(3)}$ for $k \geqslant 2$. Observe that W_{k-1} and W_k have to be in the same atom, which we denote $\mathrm{Atom}(W_{k-1})$. Thus, there is an edge e^* between w_{k-2} and v in $\mathrm{Atom}(W_{k-1})$ too (see Remark 5.2.15).

$$\begin{aligned}
\sum_{W \in \mathcal{W}_{uv}^{(3)}} \mathrm{wt}(W) &= \sum_{W \in \mathcal{W}_{uv}^{(3)}} \mathrm{wt}(W_{k-1})\mathrm{wt}(W_k)\mathrm{wt}(W_{(k-2)}) \\
&= \sum_{\substack{W' \text{ length-}(k-2) \text{ nomadic walk} \\ W'_0 = u, \\ e^* \text{ s.t. } (e^*)_1 = w_{k-2}, (e^*)_2 = v \\ \mathrm{Atom}(W'_{k-2}) \neq \mathrm{Atom}(e^*)}} \sum_{\substack{e^{(1)}, e^{(2)}: \\ \mathrm{Atom}(e^{(1)}) = \mathrm{Atom}(e^{(2)}) = \mathrm{Atom}(e^*) \\ (e^{(1)})_1 = w_{k-2}, (e^{(1)})_2 = (e^{(2)})_1, (e^{(2)})_2 = v}} \mathrm{wt}(e^{(1)})\mathrm{wt}(e^{(2)})\mathrm{wt}(W')
\end{aligned}$$

By applying Fact 5.2.11, we get

$$\begin{aligned}
&= \sum_{\substack{W' \text{ length-}(k-2) \text{ nomadic walk} \\ W'_0 = u, \\ e^* \text{ s.t. } (e^*)_1 = w_{k-2}, (e^*)_2 = v \\ \mathrm{Atom}(W'_{k-2}) \neq \mathrm{Atom}(e^*)}} (\lambda_1 + \lambda_2)\mathrm{wt}(e^*)\mathrm{wt}(W') \\
&= (\lambda_1 + \lambda_2) \sum_{W' \text{ length-}(k-1) \text{ nomadic walk from } u \text{ to } v} \mathrm{wt}(W') \\
&= (\lambda_1 + \lambda_2)p_{k-1}(A)_{uv}.
\end{aligned}$$

Now, we have for $k \geqslant 3$,

$$\begin{aligned}
\sum_{W \in \mathcal{W}_{uv}} \mathrm{wt}(W) &= \sum_{W \in \mathcal{W}_{uv}^{(1)}} \mathrm{wt}(W) + \sum_{W \in \mathcal{W}_{uv}^{(2)}} \mathrm{wt}(W) + \sum_{W \in \mathcal{W}_{uv}^{(3)}} \mathrm{wt}(W) \\
Ap_{k-1}(A)_{uv} &= \sum_{W \in \mathcal{W}_{uv}^{(1)}} \mathrm{wt}(W) + (c-1)(-\lambda_1\lambda_2)p_{k-2}(A)_{uv} + (\lambda_1 + \lambda_2)p_{k-1}(A)_{uv} \\
\sum_{W \in \mathcal{W}_{uv}^{(1)}} \mathrm{wt}(W) &= Ap_{k-1}(A)_{uv} - ((c-1)(-\lambda_1\lambda_2)p_{k-2}(A)_{uv} + (\lambda_1 + \lambda_2)p_{k-1}(A)_{uv}) \\
\sum_{W \in \mathcal{W}_{uv}^{(1)}} \mathrm{wt}(W) &= p_k(A)_{uv}.
\end{aligned}$$

For the case of $k = 2$, we carry out the above calculation by replacing $(c-1)(-\lambda_1\lambda_2)$ with $c(-\lambda_1\lambda_2)$, thus completing the inductive step. □

Generic generating functions facts. Before returning to the specifics of our problem, we give some "standard" generating function facts. These are extensions of the following simple idea: if $f(t)$ is a polynomial, then $\frac{d}{dt}\log f(t) = f'(t)\cdot f(t)^{-1}$ is (up to minor manipulations) the generating function for the power sum polynomials of its roots. We start with a general matrix version of this, which is sometimes called *Jacobi's formula* (after minor manipulations):

Proposition 5.3.3. *Let $M(t)$ be a square matrix polynomial of t. Then*

$$\frac{d}{dt}\log\det M(t) = \operatorname{tr}\big(M'(t)M(t)^{-1}\big)$$

for all $t\in\mathbb{R}$ such that $M(t)$ is invertible.

Corollary 5.3.4. *Taking $M(t) = \mathbb{1} - Ht$ for a fixed square matrix H yields*

$$\frac{d}{dt}\log\det(\mathbb{1}-Ht) = \operatorname{tr}\big(-H(\mathbb{1}-Ht)^{-1}\big) \quad\Longrightarrow\quad -t\frac{d}{dt}\log\det(\mathbb{1}-Ht) = \sum_{k\geqslant 1}\operatorname{tr}(H^k)t^k.$$

Regarding this corollary, we can derive the statement about the power sums of the roots of a polynomial $f(t)$ by taking $H = \operatorname{diag}(\lambda_1,\dots,\lambda_n)$ where the λ_i's are the roots of f. On the other hand, it actually suffices to prove Corollary 5.3.4 in the case of diagonal H, since $\det(\mathbb{1}-Ht)$ is invariant to unitary conjugation.

Growth Rate. A key term that shows up in our Ihara–Bass formula is the "growth rate" of the additive product of $\mathcal{A}$. Suppose we take t-step nomadic walk starting at a vertex v in the additive product graph, take a t-step nomadic walk back to v, and then sum over the total weight of such walks. What we get is $\big((c-1)(-\lambda_1\lambda_2)\big)^t$ (see Lemma 5.5.3 for a proof). Thus, the total weight of aforementioned walks grows exponentially in t at a rate of $(c-1)(-\lambda_1\lambda_2)$, which in this section we will refer to as α_{gr}.

The fundamental recurrence. We now relate the generating function matrix $F(t)$ to A. Using the recurrence used to generated the polynomials $p_k(x)$, one can conclude

Lemma 5.3.5. $F(t) = AF(t)t - (\lambda_1+\lambda_2)F(t)t - \alpha_{\mathrm{gr}}F(t)t^2 + (1+t\lambda_1)(1+t\lambda_2)\mathbb{1}$.

From this recurrence one may express the inverse of $F(t)$ in terms of A and c:

Corollary 5.3.6. $(1+\lambda_1 t)^{-1}(1+\lambda_2 t)^{-1}\cdot(\mathbb{1} - At + (\lambda_1+\lambda_2)t\mathbb{1} + \alpha_{\mathrm{gr}}t^2\mathbb{1})F(t) = \mathbb{1}$. *In other words, $F(t) = (1+\lambda_1 t)(1+\lambda_2 t)\mathbb{1}\cdot L(t)^{-1}$, where $L(t) \coloneqq \mathbb{1} - At + (\lambda_1+\lambda_2)t\mathbb{1} + \alpha_{\mathrm{gr}}t^2\mathbb{1}$ is the "deformed Laplacian" appearing in the statement of our Ihara–Bass theorem.*

Strategy for the rest of the proof. The strategy will be to apply Proposition 5.3.3 with the deformed Laplacian $L(t)$. On the left side we'll get a determinant involving A. On the right side we'll get a trace involving $L(t)^{-1}$, which is essentially $F(t)$. In turn, $\operatorname{tr}(F(t))$ is a generating function for nomadic closed walks, which we can hope to relate to B (although there will be an edge case to deal with).

Let's begin executing this strategy. By Proposition 5.3.3 we have

$$-t\frac{d}{dt}\log\det L(t) = -t\cdot\operatorname{tr}\big(L'(t)L(t)^{-1}\big)$$

$$= -t \cdot \operatorname{tr}\big((\mathbb{1}(\lambda_1+\lambda_2) - A + 2\alpha_{\mathrm{gr}} t\mathbb{1}) \cdot ((1+\lambda_1 t)(1+\lambda_2 t))^{-1} F(t)\big)$$
$$= \frac{1}{(1+\lambda_1 t)(1+\lambda_2 t)} \operatorname{tr}\big(-(\lambda_1+\lambda_2)F(t)t + AF(t)t - 2\alpha_{\mathrm{gr}} F(t)t^2\big)$$

where we used Corollary 5.3.6. Now using Lemma 5.3.5 again we may infer

$$-(\lambda_1+\lambda_2)F(t)t + AF(t)t - 2\alpha_{\mathrm{gr}} F(t)t^2 = (1-\alpha_{\mathrm{gr}} t^2)F(t) - (1+\lambda_1 t)(1+\lambda_2 t)\mathbb{1};$$

combining the previous two identities yields

$$-t\frac{d}{dt}\log\det L(t) = \operatorname{tr}\left(\frac{1-\alpha_{\mathrm{gr}} t^2}{(1+\lambda_1 t)(1+\lambda_2 t)} F(t) - \mathbb{1}\right). \tag{5.3}$$

Nomadic walks. The right side above is $\operatorname{tr}(F(t))$ up to some scaling/translating. By definition, $\operatorname{tr}(F(t))$ is the generating function for nomadic *circuits* (closed walks) with any starting point. A first instinct is therefore to expect that

$$\operatorname{tr}(F(t)) \stackrel{?}{=} \sum_{k\geqslant 0} \operatorname{tr}(B^k) t^k, \tag{5.4}$$

as $\operatorname{tr}(B^k)$ is the weight of closed length-k circuits of direct edges in the nomadic world. However this is not quite right: $\operatorname{tr}(B^k)$ only weighs the nomadic circuits whose first and last edge are not in the same atom. The nomadic circuits that are not weighed can be identified either as (i) "tailed" nomadic circuits, i.e., those where the last directed edge is the reverse of the first directed edge; (ii) "stretched" nomadic circuits, i.e., those where the last directed edge is distinct from but in the same atom as the first directed edge. E.g., $\operatorname{tr}(B^k)$ would fail to count the following:

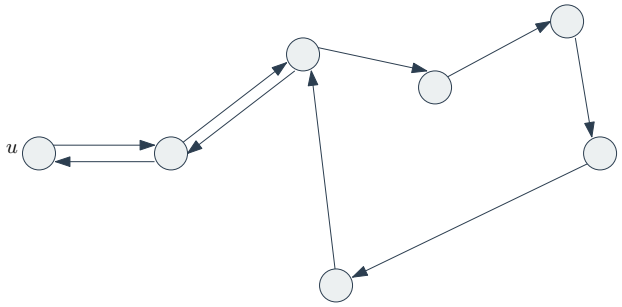

Figure 5.8: A length-9 nomadic walk from u to u with a *tail* of length 2

Thus we need to correct (5.4).

Definition 5.3.7. With the $-\mathbb{1}$ taking care of the omission of $k=0$, we define

$$\mathrm{Tails}(t) = \sum_{k\geqslant 1} (\text{weight of nomadic circuits of length } k) t^k = \operatorname{tr}(F(t) - \mathbb{1}). \tag{5.5}$$

We also define

$$\mathrm{NoTails}(t) = \sum_{k\geqslant 1} (\text{weight of tail-less nomadic circuits of length } k) t^k$$

and

$$\mathrm{Simple}(t) = \sum_{k\geqslant 1}(\text{weight of non-stretched, tail-less nomadic circuits of length } k)t^k$$
$$= \sum_{k\geqslant 1}\mathrm{tr}(B^k)t^k = -t\frac{d}{dt}\log\det(\mathbb{1} - Bt), \quad (5.6)$$

where the last equality used Corollary 5.3.4.

Tails vs. no tails vs. simple: more generating functions. We finish by relating $\mathrm{Tails}(t)$, $\mathrm{NoTails}(t)$ and $\mathrm{Simple}(t)$. This is the recipe:

> A general nomadic circuit of length k is constructed from a tail-less nomadic circuit of length $k-2\ell$ with a tail of length-ℓ attached to one of its vertices.

Tail-less nomadic circuits can be classified as (i) non-stretched tail-less nomadic circuits, and (ii) stretched, tail-less nomadic circuits, for which,

$$\mathrm{NoTails}(t) - \mathrm{Simple}(t) = \sum_{k\geqslant 1}(\text{weight of stretched, tail-less nomadic walks of length } k)t^k.$$

Consider a stretched, tail-less nomadic walk of length k that starts at vertex v, takes the edge e from v to u, goes on a nomadic walk W from u to w, and finally takes edge e' from w to v to end the walk at v. Note that e and e' are part of the same atom A_i. Summing over all v in atom A_i and applying Fact 5.2.11 gives

$$\sum_{v\sim A_i}\mathrm{wt}(A_i(v,u))\mathrm{wt}(A_i(w,v))\mathrm{wt}(W) = (\lambda_1+\lambda_2)\mathrm{wt}(A_i(w,u))\mathrm{wt}(W) = (\lambda_1+\lambda_2)\mathrm{wt}(W')$$

where W' is a nomadic circuit of length $k-1$ that starts at w, takes edge $A_i(w,u)$ in the first step, and then takes walk W. From this, we derive

$$\mathrm{NoTails}(t) - \mathrm{Simple}(t) = (\lambda_1+\lambda_2)t\cdot\mathrm{Simple}(t).$$

It's easy to count the total weight of tails of length ℓ one can attach to a given vertex of a tail-less nomadic circuit: if the tail-less nomadic circuit is non-stretched, the first edge can be chosen by picking any edge in $(c-2)$ atoms and each of the remaining $\ell-1$ edges can be chosen by picking any edge $(c-1)$ atoms; and if the tail-less nomadic circuit is stretched, each edge (including the first one) can be chosen anywhere from $(c-1)$ atoms. From this it's easy to derive

$$\begin{aligned}\mathrm{Tails}(t) &= \big(1+(-\lambda_1\lambda_2)(c-2)t^2 + (-\lambda_1\lambda_2)^2(c-2)(c-1)t^4+\cdots\big)\mathrm{Simple}(t)\\ &\quad+\big(1+(-\lambda_1\lambda_2)(c-1)t^2+(-\lambda_1\lambda_2)^2(c-1)^2t^4+\cdots\big)(\mathrm{NoTails}(t)-\mathrm{Simple}(t))\\ &= \frac{1-(-\lambda_1\lambda_2)t^2}{1-(c-1)(-\lambda_1\lambda_2)t^2}\mathrm{Simple}(t) + \frac{(\lambda_1+\lambda_2)t}{1-(c-1)(-\lambda_1\lambda_2)t^2}\mathrm{Simple}(t)\\ &\Leftrightarrow \mathrm{Simple}(t) = \frac{1-\alpha_{\mathrm{gr}}t^2}{(1+\lambda_1t)(1+\lambda_2t)}\mathrm{Tails}(t). \quad (5.7)\end{aligned}$$

Using $\mathrm{Tails}(t) = \mathrm{tr}(F(t)-\mathbb{1})$ (i.e., (5.5)), we obtain:

Corollary 5.3.8. $\text{Simple}(t) = \text{tr}\left(\frac{1-\alpha_{\text{gr}}t^2}{(1+\lambda_1 t)(1+\lambda_2 t)}(F(t)-\mathbb{1})\right)$.

But this is *almost* the same as (5.3). The difference is

$$\begin{aligned}\text{tr}\left(\mathbb{1} - \frac{1-\alpha_{\text{gr}}t^2}{(1+\lambda_1 t)(1+\lambda_2 t)}\mathbb{1}\right) &= \text{tr}\left(\frac{(\lambda_1+\lambda_2)t + (c-2)(-\lambda_1\lambda_2)t^2}{(1+\lambda_1 t)(1+\lambda_2 t)}\mathbb{1}\right) \\ &= |V| \cdot \frac{(\lambda_1+\lambda_2)t + (c-2)(-\lambda_1\lambda_2)t^2}{(1+\lambda_1 t)(1+\lambda_2 t)}.\end{aligned}$$

Combining the above with (5.3), Corollary 5.3.8, and (5.6), we finally conclude

$$-t\frac{d}{dt}\log\det L(t) + |V| \cdot \frac{(\lambda_1+\lambda_2)t + (c-2)(-\lambda_1\lambda_2)t^2}{(1+\lambda_1 t)(1+\lambda_2 t)} = -t\frac{d}{dt}\log\det(\mathbb{1}-Bt).$$

Finally, dividing by $-t$, integrating (which leaves an unspecified additive constant), and exponentiating (now there is an unspecified multiplicative constant) yields

$$(\text{const.}) \cdot (1+\lambda_1 t)^{|V|\frac{c\lambda_2}{\lambda_2-\lambda_1}-1}(1+\lambda_2 t)^{|V|\frac{c\lambda_1}{\lambda_1-\lambda_2}-1}\det L(t) = \det(\mathbb{1}-Bt).$$

By consideration of $t=0$ we see that the constant must be 1.

5.4 Connecting the adjacency and nomadic spectrum

Let $\mathcal{A} = (A_1, \ldots, A_c)$ be a sequence of atoms with two distinct eigenvalues λ_1 and λ_2, let $\mathcal{H}$ be an r-ary, c-atom constraint graph, and let $\mathcal{I} = \mathcal{A}(\mathcal{H})$ be the corresponding instance graph. We use A for the adjacency matrix of $\mathcal{I}$, B for its nomadic walk matrix, V for its vertex set, and E for its edge set. Recall that α_{gr} is defined as $(c-1)(-\lambda_1\lambda_2)$.

We want to use Theorem 5.3.1 to describe the spectrum of B with respect to that of A. We will refer to eigenvalues of B with the letter μ and eigenvalues of A with the letter ν.

First, notice that if t is such that $\det(\mathbb{1}-Bt) = 0$, then $\mu = 1/t$ has $\det(\mu\mathbb{1}-B) = 0$, meaning μ is an eigenvalue of B. Thus we want to find for which values of t does the left-hand side of the expression in Theorem 5.1.4 become 0 in order to deduce the spectrum of B.

It is easy to see that when $t = -1/\lambda_1$ and $t = -1/\lambda_2$ the left-hand side is always 0, so $-\lambda_1$ is an eigenvalue of B with multiplicity $|V|(\frac{c\lambda_2}{\lambda_2-\lambda_1} - 1)$ and $-\lambda_2$ is an eigenvalue with multiplicity $|V|(\frac{c\lambda_1}{\lambda_1-\lambda_2} - 1)$. The remaining eigenvalues are given by the values of t for which $\det(L(t)) = 0$. Let t be such that $\det(L(t)) = 0$; then we have that $L(t)$ is non-invertible, which means there is some vector v in the nullspace of $L(t)$. By rearranging the equality $L(t)v = 0$ we get:

$$Av = \frac{1+(\lambda_1+\lambda_2)t+\alpha_{\text{gr}}t^2}{t}v.$$

This implies that $\frac{1+(\lambda_1+\lambda_2)t+\alpha_{\text{gr}}t^2}{t}$ is an eigenvalue of A. Let ν be some eigenvalue of A; then we have that $\nu = \frac{1+(\lambda_1+\lambda_2)t+\alpha_{\text{gr}}t^2}{t}$ for some t. If we rearrange the previous expression we get the following quadratic equation in t:

$$1 + (\lambda_1 + \lambda_2 - \nu)t + \alpha_{\text{gr}} t^2 = 0.$$

By solving this expression for t and then using the fact that $\mu = 1/t$ we get (notice that $c > 1$ and $\lambda_1 \lambda_2 \neq 0$):

$$\mu = \frac{-2\alpha_{\text{gr}}}{\lambda_1 + \lambda_2 - \nu \pm \sqrt{(\lambda_1 + \lambda_2 - \nu)^2 - 4\alpha_{\text{gr}}}}.$$

To analyze the previous we look at three cases:

1. $\nu > \lambda_1 + \lambda_2 + 2\sqrt{\alpha_{\text{gr}}}$. In this case the discriminant is always positive. If we look at the $-$ branch of the $\pm$ we further get that the denominator of the previous formula is always less than $-2\sqrt{\alpha_{\text{gr}}}$ which means we have that μ is real and $\mu > \sqrt{\alpha_{\text{gr}}}$. Additionally, we have that in this interval μ is an increasing function of ν.
2. $\nu < \lambda_1 + \lambda_2 - 2\sqrt{\alpha_{\text{gr}}}$. This is analogous to the previous case; if we look at the $+$ branch we have that μ is real and $\mu < -\sqrt{\alpha_{\text{gr}}}$. Additionally, we have that in this interval μ is a decreasing function of ν.
3. $\nu \in [\lambda_1 + \lambda_2 - 2\sqrt{\alpha_{\text{gr}}}, \lambda_1 + \lambda_2 + 2\sqrt{\alpha_{\text{gr}}}]$, for each such ν we get a pair of anti-conjugate complex numbers, meaning a pair $x, \overline{x}$ such that $x\overline{x} = -1$.

Finally, the spectrum of B also contains 0 with multiplicity $2|E| - |V| \left(2 + (\frac{c\lambda_1}{\lambda_1 - \lambda_2} - 1) + (\frac{c\lambda_2}{\lambda_2 - \lambda_1} - 1)\right)$, which we get because the degrees of the polynomials in the left-hand side and right-hand do not match; the right-hand side has degree $2|E|$ but we only described $|V| \left(2 + (\frac{c\lambda_1}{\lambda_1 - \lambda_2} - 1) + (\frac{c\lambda_2}{\lambda_2 - \lambda_1} - 1)\right)$ roots.

We can now summarize the eigenvalues of B in the following way:

- $-\lambda_1$ with multiplicity $|V|(\frac{c\lambda_2}{\lambda_2 - \lambda_1} - 1)$;
- $-\lambda_2$ with multiplicity $|V|(\frac{c\lambda_1}{\lambda_1 - \lambda_2} - 1)$;
- for each eigenvalue ν of A we get two eigenvalues that are solutions to the previous quadratic equation;
- 0 with multiplicity $2|E| - |V| \left(2 + (\frac{c\lambda_1}{\lambda_1 - \lambda_2} - 1) + (\frac{c\lambda_2}{\lambda_2 - \lambda_1} - 1)\right)$;

The distribution of the eigenvalues that come from A forms a sort of semicircle. To showcase this behavior we display an example of the spectrum of typical lifted instance in Figure 5.9.

We can now prove the central theorem of this section:

Theorem 5.4.1. *Let $\boldsymbol{\mathcal{I}}_n$ be a random additive n-lift of $\mathcal{A}$ with adjacency matrix $A_{\boldsymbol{\mathcal{I}}_n}$, and let $\epsilon > 0$. Then:*

$$\mathbf{Pr}\left[\rho(A_{\boldsymbol{\mathcal{I}}_n}) \in [\lambda_1 + \lambda_2 - 2\sqrt{\alpha_{\text{gr}}} - \varepsilon, \lambda_1 + \lambda_2 + 2\sqrt{\alpha_{\text{gr}}} + \varepsilon\right] = 1 - o_n(1)$$

Proof. First recall Theorem 5.1.6 (for fully formal statement, see Theorem 5.6.20) and notice that $\rho(|B|) = \alpha_{\text{gr}}$, which follows by using the trivial upper bound of α_{gr}^{2k} on $\operatorname{tr}\left(|B|^k \left(|B|^*\right)^k\right)$. From cases 1 and 2 in the previous analysis we get that if $\rho(A_{\boldsymbol{\mathcal{I}}_n}) \notin [\lambda_1 + \lambda_2 - 2\sqrt{\alpha_{\text{gr}}} - \varepsilon, \lambda_1 + \lambda_2 + 2\sqrt{\alpha_{\text{gr}}} + \varepsilon]$ there is some constant δ such that $\rho(B_n) > \sqrt{\alpha_{\text{gr}}} + \delta$, which happens with $o_{n\to\infty}(1)$ probability by Theorem 5.6.20. □

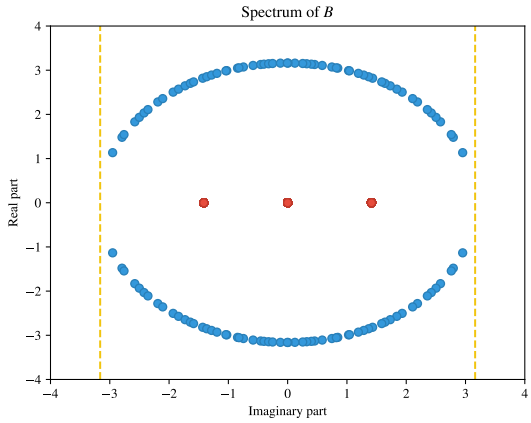

Figure 5.9: The spectrum of B for a additive 15-lift of 6 copies of a Sort_4 graph. The blue dots are eigenvalues that come from eigenvalues of A, the red dots are either $-\lambda_1$, $-\lambda_2$ or 0 and the yellow line is the limit $\sqrt{\alpha_{\mathrm{gr}}}$.

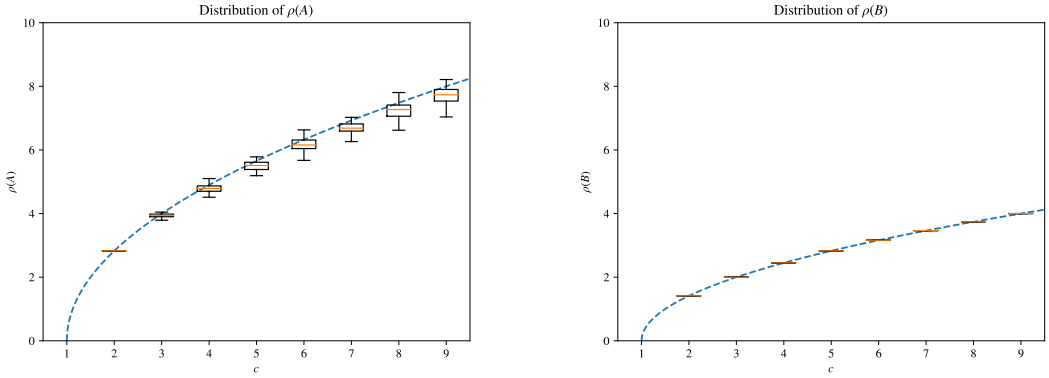

Figure 5.10: A box plot of $\rho(A)$ and $\rho(B)$ of 100 samples of random instance graphs as a function of c with $n = 15$, $r = 4$ and all atoms are the Sort_4 graph. The dashed line shows the theoretical bound prediction of $2\sqrt{\alpha_{\mathrm{gr}}}$ for A and $\sqrt{\alpha_{\mathrm{gr}}}$ for B.

Also, we note that even though throughout our proof we hide various constant factors, the bounds obtained in Theorem 5.4.1 and Theorem 5.6.20 are empirically visible for very small values of n and c. To justify this claim we show in Figure 5.10 a plot of samples of random instance graphs for different values of c with a fixed small n.

5.5 Additive products of 2-eigenvalue atoms

In this section, we let $\mathcal{A} = (A_1, \ldots, A_c)$ be a sequence of $\{\pm 1\}$-*weighted* atoms with the same pair of exactly two distinct eigenvalues, λ_1 and λ_2. We also let $X \coloneqq A_1 \dotplus \cdots \dotplus A_c$ be the additive product graph. We use A_X to denote the adjacency operator of X. In this section, $\boldsymbol{\mathcal{I}}_n$ is the instance graph of a random additive n-lift of $\mathcal{A}$ with negations, and we use $A_{\boldsymbol{\mathcal{I}}_n}$ to denote its

adjacency matrix. Finally, we recall $\alpha_{\mathrm{gr}} := (c-1)(-\lambda_1\lambda_2)$ and define the quantity $r_X := 2\sqrt{\alpha_{\mathrm{gr}}}$.

The main results that this section is dedicated to proving are:

Theorem 5.5.1. *The following are true about the spectrum of X:*

1. $\mathrm{Spec}(A_X) \subseteq [\lambda_1 + \lambda_2 - r_X, \lambda_1 + \lambda_2 + r_X]$;
2. $\lambda_1 + \lambda_2 - r_X$ *and* $\lambda_1 + \lambda_2 + r_X$ *are both in* $\mathrm{Spec}(A_X)$.

Theorem 5.5.2. *For every $\varepsilon > 0$, for large enough n, there are $|V(\boldsymbol{\mathcal{I}}_n)| \times |V(\boldsymbol{\mathcal{I}}_n)|$ positive semidefinite matrices M_+ and M_- with all-ones diagonals such that*

$$\langle A_{\boldsymbol{\mathcal{I}}_n}, M_+\rangle \geqslant (\lambda_1 + \lambda_2 + r_X - \varepsilon)n$$
$$\langle A_{\boldsymbol{\mathcal{I}}_n}, M_-\rangle \leqslant (\lambda_1 + \lambda_2 - r_X + \varepsilon)n.$$

with probability $1 - o_n(1)$.

In this section, when we measure the distance between vertices u and v in an instance graph $\boldsymbol{\mathcal{I}}_n$, we look at the corresponding vertices in the constraint graph $\boldsymbol{\mathcal{H}}$, and define $d(u,v) := \frac{d_{\boldsymbol{\mathcal{K}}}(u,v)}{2}$. We use $\mathcal{P}_{uv}$ to refer to the collection of edges comprising the shortest path between u and v. We begin with a statement about the 'growth rate' of X.

Lemma 5.5.3. *For all vertices v in $V(X)$, for $t \geqslant 1$ we have*

$$\sum_{u:d(u,v)=t} \prod_{\{i,j\}\in\mathcal{P}_{uv}} (A_X)_{ij}^2 = c(c-1)^{t-1}(-\lambda_1\lambda_2)^t.$$

Proof. We proceed by induction. When $t = 1$, the statement immediately follows from Fact 5.2.10. Suppose the equality is true for some $t = \ell - 1$, we will show how statement follows for $t = \ell$.

$$\sum_{u:d(u,v)=\ell} \prod_{\{i,j\}\in\mathcal{P}_{uv}} (A_X)_{ij}^2 = \sum_{u:d(u,v)=\ell-1} \left(\prod_{\{i,j\}\in\mathcal{P}_{uv}} (A_X)_{ij}^2\right) \cdot \left(\sum_{\substack{u'\in N(u)\\ d(u',v)=\ell}} (A_X)_{uu'}^2\right)$$

From Fact 5.2.10, $\sum_{\substack{u'\sim u\\ d(u',v)=t}} (A_X)_{uu'}^2$ is equal to $(c-1)(-\lambda_1\lambda_2)$, which means the above is equal to

$$\begin{aligned}
&= \sum_{u:d(u,v)=\ell-1} \left(\prod_{\{i,j\}\in\mathcal{P}_{uv}} (A_X)_{ij}^2\right)(c-1)(-\lambda_1\lambda_2)\\
&= (c-1)^{\ell-2}c(-\lambda_1\lambda_2)^{\ell-1}(c-1)(-\lambda_1\lambda_2)\\
&= c(c-1)^{\ell-1}(-\lambda_1\lambda_2)^{\ell}.
\end{aligned}$$

□

Corollary 5.5.4. *Since all the weights of X are $\{\pm 1\}$-valued, the degree of every vertex in X equals $c(-\lambda_1\lambda_2)$.*

5.5.1 Enclosing the spectrum

Let B_X denote the nomadic walk operator of X. In this section, we show

$$\operatorname{Spec}(A_X) \subseteq [\lambda_1 + \lambda_2 - r_X, \lambda_1 + \lambda_2 + r_X]\,.$$

The first part of the proof will involve showing that the spectral radius of B_X is bounded by $\sqrt{\alpha_{\mathrm{gr}}}$, and the second part translates this bound to the desired one on $\operatorname{Spec}(A_X)$. Both these components closely follow proofs from the work of Angel et al.; the former after [AFH15, Theorem 4.2] and the latter after [AFH15, Theorem 1.5].

Lemma 5.5.5. $\operatorname{Spec}(B_X) \subseteq \left[-\sqrt{\alpha_{\mathrm{gr}}}, \sqrt{\alpha_{\mathrm{gr}}}\right]$.

Proof. Arbitrarily fix a root r of X. Recall that the spectral radius of B_X is equal to $\lim \left(\|B_X^k\|_{\mathrm{op}}\right)^{1/k}$, and hence it suffices to bound $\left|\langle g, B_X^k f\rangle\right|$ for arbitrary f and g with $\|f\| = \|g\| = 1$.

We can decompose every nomadic walk of length k into two segments, a segment of i steps towards r followed by a sequence of $k - i$ steps away from r; henceforth, we call length-k nomadic walks with such a decomposition (i, k)-nomadic walks. For every pair of directed edges e and e' such that $e, e_1, \ldots, e_{k-1}, e'$ is an (i, k)-nomadic walk, let $a(e, e') := \alpha_{\mathrm{gr}}^{k/2-i}$. From Lemma 5.5.3, the number of (i, k)-nomadic walks starting at a fixed e is at most $\frac{c}{c-1}\alpha_{\mathrm{gr}}^{k-i}$. Similarly, the number of (i, k)-nomadic walks ending at fixed e' is at most $\frac{c}{c-1}\alpha_{\mathrm{gr}}^{i}$. Now, we are ready to bound $\left|\langle g, B_X^k f\rangle\right|$ by imitating the proof of [AFH15, Theorem 4.2].

$$\begin{aligned}
\left|\langle g, B_X^k f\rangle\right| &\leqslant \left| \sum_{e,e_1,\ldots,e_{k-1},e' \text{ nomadic}} f(e')g(e) \right| \\
&\leqslant \sum_{e,e_1,\ldots,e_{k-1},e' \text{ nomadic}} |f(e')g(e)| \\
&\leqslant \sum_{e,e_1,\ldots,e_{k-1},e' \text{ nomadic}} a(e,e')f(e')^2 + \frac{1}{a(e,e')}g(e)^2 \\
&\leqslant \sup_{e'} \left(\sum_{e,e_1,\ldots,e_{k-1},e' \text{ nomadic}} a(e,e') \right) \|f\|_2^2 + \sup_{e} \left(\sum_{e,e_1,\ldots,e_{k-1},e' \text{ nomadic}} \frac{1}{a(e,e')} \right) \|g\|_2^2 \\
&\leqslant \sum_{i=0}^{k} \sup_{e'} \left(\sum_{(i,k)\text{-nomadic walks ending at } e'} a(e,e') \right) + \sup_{e} \left(\sum_{(i,k)\text{-nomadic walks starting at } e} \frac{1}{a(e,e')} \right) \\
&\leqslant \sum_{i=0}^{k} \alpha_{\mathrm{gr}}^{k/2-i} \cdot \frac{c}{c-1}\alpha_{\mathrm{gr}}^{i} + \sum_{i=0}^{k} \alpha_{\mathrm{gr}}^{i-k/2} \cdot \frac{c}{c-1}\alpha_{\mathrm{gr}}^{k-i} \\
&= \frac{2kc}{c-1}\alpha_{\mathrm{gr}}^{k/2}
\end{aligned}$$

Thus, we have

$$\|B_X^k\|_{\mathrm{op}} \leqslant \frac{2kc}{c-1}\alpha_{\mathrm{gr}}^{k/2}$$

and taking the limit of $\|B_X^k\|_{\mathrm{op}}^{1/k}$ for k approaching infinity yields the desired statement. □

Lemma 5.5.6. *If 0 is an approximate eigenvalue of $Q_t \coloneqq (t^2+(c-1)(-\lambda_1\lambda_2))\mathbb{1} - A_X t + (\lambda_1+\lambda_2)\mathbb{1}t$, then it is also an approximate eigenvalue of $B_X - t\mathbb{1}$ as long as $t \neq -\lambda_1, -\lambda_2$.*

Proof. Let f be an ε-approximate eigenfunction of unit norm of Q_t, then we construct a $C\varepsilon$-approximate eigenfunction g of $B_X - t\mathbb{1}$ defined on pairs uv such that u and v are incident to a common atom for an absolute constant $C > 0$ as follows,

$$g_{uv} \coloneqq \left(\sum_{w:\{v,w\}\in \mathrm{Atom}(\{u,v\})} (A_X)_{vw} f_w \right) - (\lambda_1+\lambda_2+t) f_v$$

for every edge $\{u, v\}$ of X.

$$\begin{aligned}
((B_X - t\mathbb{1})g)_{uv} &= \left(\sum_{\substack{w:\\ \{v,w\}\notin \mathrm{Atom}(\{u,v\})}} (B_X)_{uv,vw} g_{vw} \right) - t g_{uv} \\
&= \left(\sum_{\substack{w:\\ \{v,w\}\notin \mathrm{Atom}(\{u,v)\}}} (A_X)_{vw} \left(\sum_{\substack{x:\\ \{w,x\}\in \mathrm{Atom}(\{v,w\})}} (A_X)_{wx} f_x - (\lambda_1+\lambda_2+t) f_w \right) \right) - t g_{uv} \\
&= \left(\sum_{\substack{w:\\ \{v,w\}\notin \mathrm{Atom}(\{u,v)\}}} \sum_{\substack{x:\\ \{w,x\}\in \mathrm{Atom}(\{v,w\})}} (A_X)_{vw}(A_X)_{wx} f_x \right) - \\
&\quad \left(\sum_{\substack{w:\\ \{v,w\}\notin \mathrm{Atom}(\{u,v)\}}} (\lambda_1+\lambda_2+t)(A_X)_{vw} f_w \right) - t g_{uv}
\end{aligned}$$

Using Fact 5.2.10 and Fact 5.2.11, the first term of the three above can be rewritten as

$$(c-1)(-\lambda_1\lambda_2) f_v + (\lambda_1+\lambda_2) \sum_{w:\{v,w\}\notin \mathrm{Atom}(\{u,v\})} (A_X)_{vw} f_w$$

which lets us continue the chain of equalities

$$\begin{aligned}
&= (c-1)(-\lambda_1\lambda_2) f_v - t \sum_{\substack{w:\\ \{v,w\}\notin \mathrm{Atom}(\{u,v)\}}} (A_X)_{vw} f_w \\
&\quad - t \left(\sum_{w:\{v,w\}\in \mathrm{Atom}(\{u,v\})} (A_X)_{vw} f_w \right) + t(\lambda_1+\lambda_2+t) f_v \\
&= (c-1)(-\lambda_1\lambda_2) f_v - t(Af)_v + t(\lambda_1+\lambda_2+t) f_v \\
&= (Q_t f)_v.
\end{aligned}$$

Thus,

$$\|(B_X - t\mathbb{1})g\|_2^2 = \sum_{\{u,v\}\in E(X)} ((B_X - t\mathbb{1})g)_{uv}^2 + ((B_X - t\mathbb{1})g)_{vu}^2 = d\sum_{v\in V} (Q_t f)_v^2 \leqslant d\varepsilon^2$$

It remains to show that the norm of g is bounded from above and below. Fix a vertex u and an atom $\widetilde{A}$ incident to u. Consider $g^{(u,\widetilde{A})}$, the restriction of g to entries uv such that the edge $\{u,v\}$ is in $\widetilde{A}$, and $f^{(\widetilde{A})}$, the restriction of f to vertices v such that $\widetilde{A}$ is incident to v. Observe that $g^{(u,\widetilde{A})} = (A_{\widetilde{A}} - (\lambda_1+\lambda_2+t)\mathbb{1})f^{(\widetilde{A})}$. Since the min eigenvalue of $A_{\widetilde{A}} - (\lambda_1+\lambda_2+t)\mathbb{1}$ is nonzero as long as $t \neq -\lambda_1, -\lambda_2$, the ℓ_2 norm of g is bounded from below. To prove that the ℓ_2 norm of g is bounded from above, observe that

$$\begin{aligned}\|g\|_2^2 &= \sum_{\widetilde{A}\in\mathrm{Atoms}(X)} \sum_{(u,v):\{u,v\}\in\widetilde{A}} \left(\left(\sum_{w:\{v,w\}\in\widetilde{A}} (A_X)_{vw} f_w \right) - (\lambda_1+\lambda_2+t) f_v \right)^2 \\ &\leqslant 2 \sum_{\widetilde{A}\in\mathrm{Atoms}(X)} \sum_{(u,v):\{u,v\}\in\widetilde{A}} \left(\sum_{\{v,w\}\in\widetilde{A}} (A_X)_{vw}^2 f_w^2 + (\lambda_1+\lambda_2+t)^2 f_v^2 \right)\end{aligned}$$

There is some coefficient α such that the weight on f_v^2 for each v in the above sum is bounded by α, thereby giving a bound of

$$2\sum_{v\in V} \alpha f_v^2 \leqslant 2\alpha \|f\|_2^2 \leqslant 2\alpha. \qquad \square$$

Proof of Item 1 in Theorem 5.5.1. Let Q_t be as defined in the statement of Lemma 5.5.6. It can be verified that 0 is an approximate eigenvalue of either $Q_{-\lambda_1}$ or $Q_{-\lambda_2}$ if and only if $d_X := c(-\lambda_1\lambda_2)$, which we recall from Corollary 5.5.4 is the degree of every vertex in X, is in the spectrum of A_X. Let $\mu_+ := \lambda_1+\lambda_2+r_X+\eta$ be in spectrum of A_X. If $\mu_+ \neq d_X$, then we can conclude from Lemma 5.5.6 that

$$\alpha_{\mathrm{gr}} + \eta + \sqrt{\eta\alpha_{\mathrm{gr}} + \eta^2/4}$$

is an approximate eigenvalue of B_X. Since $\mathrm{Spec}(B_X)$ is contained in $[-\sqrt{\alpha_{\mathrm{gr}}}, \sqrt{gr}]$, η cannot be positive. A similar argument applied to $\mu_- := \lambda_1+\lambda_2-r_X-\eta$ precludes η from being positive as long as $\mu_- \neq d_X$. As a result, we can conclude that $\mathrm{Spec}(A_X)$ is contained in $[\mu_-, \mu_+] \cup \{d_X\}$. If d_X is in the interval $[\mu_-, \mu_+]$, then we are done. If not, then it remains to show that d_X is not in $\mathrm{Spec}(A_X)$. Since X is $\{\pm 1\}$-weighted and the degree of each vertex is d_X, any nonzero x satisfying $A_X x = d_X x$ must have the same nonzero magnitude in all its entries. However, such x has unbounded ℓ_2 norm, and hence A_X has no eigenvectors with eigenvalue d_X in $\ell_2(V)$. If d_X is in $\mathrm{Spec}(A_X)$, it is an isolated point in the spectrum, and hence, by Theorem 5.2.25, is an eigenvalue of A_X, which means d_X cannot be in $\mathrm{Spec}(A_X)$. $\square$

5.5.2 Construction of Witness Vectors

Lemma 5.5.7 (Item 2 of Theorem 5.5.1 restated)**.** *There exists $\lambda_- \leqslant \lambda_1+\lambda_2-r_X$ and $\lambda_+ \geqslant \lambda_1+\lambda_2+r_X$ in the spectrum of A_X.*

Proof. Let $\delta > 0$ be a parameter to be chosen later. First define ρ as

$$\rho(s) := \frac{s(1-\delta)}{\sqrt{(c-1)(-\lambda_1\lambda_2)}}$$

Then, for vertex v and define $f_v^{(s)}$ in the following way.

$$f_v^{(s)}(u) := \rho(s)^{d(u,v)} \prod_{\{i,j\}\in\mathcal{P}_{uv}} (A_X)_{ij} \quad \text{where } \mathcal{P}_{uv} \text{ is the unique nomadic walk between } u \text{ and } v \tag{5.8}$$

To show the lemma, it suffices to prove the claim that for every $\varepsilon > 0$, there is suitable choice of δ so that

$$\frac{\langle f_v^{(-1)}, A_X f_v^{(-1)}\rangle}{\langle f_v^{(-1)}, f_v^{(-1)}\rangle} < \lambda_1 + \lambda_2 - r_X + \varepsilon$$

and

$$\frac{\langle f_v^{(1)}, A_X f_v^{(1)}\rangle}{\langle f_v^{(1)}, f_v^{(1)}\rangle} > \lambda_1 + \lambda_2 + r_X - \varepsilon$$

We proceed by analyzing the expression $\langle f_v^{(s)}, A_X f_v^{(s)}\rangle$.

$$\begin{aligned}
\langle f_v^{(s)}, A_X f_v^{(s)}\rangle &= \sum_{u\in V} f_v^{(s)}(u) A_X f_v^{(s)}(u) \\
&= f_v^{(s)}(v) \sum_{w\in N(v)} (A_X)_{vw} f_v^{(s)}(w) + \sum_{u\in V, u\neq v} f_v^{(s)}(u) \sum_{w\in N(u)} (A_X)_{uw} f_v^{(s)}(w) \\
&= \sum_{w\in N(v)} (A_X)_{vw}^2 \rho(s) + \sum_{u\in V, u\neq v} f_v^{(s)}(u) \sum_{w\in N(u)} (A_X)_{uw} f_v^{(s)}(w) \qquad (5.9)
\end{aligned}$$

Let $w_0, w_1, \ldots w_{T-1}, w_T$ be the sequence of vertices from the unique nomadic walk between u and v where $w_0 = u$ and $w_T = v$. Now, let $u^* = w_1$. Recall the notation $\mathcal{P}_{u,v}$ used to denote the unique nomadic walk between u and v as a sequence of edges. Let $W_{u,v} := \rho(s)^{d(u,v)} \prod_{\{i,j\}\in\mathcal{P}_{u,v}} (A_X)_{ij}$. Using the notation we just developed, along with applying Fact 5.2.10 on the first term of the above, we get

$$\begin{aligned}
(5.9) &= c(-\lambda_1\lambda_2)\rho(s) + \sum_{u\in V, u\neq v} \rho(s) W_{u^*v} (A_X)_{uu^*} \cdot \\
&\left((A_X)_{uu^*} W_{u^*v} + \sum_{w\in \mathrm{Atom}(\{u^*,u\})} \rho(s)(A_X)_{u^*w}(A_X)_{wu} W_{u^*v} + \sum_{\substack{w\notin \mathrm{Atom}(\{u,u^*\}) \\ w\in N(u)}} \rho(s)^2 (A_X)_{u^*u} (A_X)_{uw}^2 W_{u^*v} \right) \\
&= c(-\lambda_1\lambda_2)\rho(s) + \sum_{u\in V, u\neq v} \rho(s) W_{u^*v}^2 (A_X)_{uu^*}^2 \cdot
\end{aligned}$$

$$\left(1+\frac{\sum_{w\in\mathrm{Atom}(\{u^*,u\})}\rho(s)(A_X)_{u^*w}(A_X)_{wu}}{A_{uu^*}}+\sum_{\substack{w\notin\mathrm{Atom}(\{u,u^*\})\\ w\in N(u)}}(A_X)_{uw}^2\rho(s)^2\right)$$

Now we apply Fact 5.2.10 and Fact 5.2.11 and get

$$\begin{aligned}
&= c(-\lambda_1\lambda_2)\rho(s)+\sum_{u\in V,u\neq v}\rho(s)W_{u^*v}^2(A_X)_{uu^*}^2\cdot\left(1+\rho(s)(\lambda_1+\lambda_2)+(c-1)(-\lambda_1\lambda_2)\rho(s)^2\right)\\
&= c(-\lambda_1\lambda_2)\rho(s)+\sum_{u\in V,u\neq v}W_{uv}^2\cdot\frac{1+\rho(s)(\lambda_1+\lambda_2)+(c-1)(-\lambda_1\lambda_2)\rho(s)^2}{\rho(s)}\\
&= c(-\lambda_1\lambda_2)\rho(s)+\left(\|f_v^{(s)}\|^2-1\right)\cdot\frac{1+\rho(s)(\lambda_1+\lambda_2)+(c-1)(-\lambda_1\lambda_2)\rho(s)^2}{\rho(s)}\\
&= c(-\lambda_1\lambda_2)\rho(s)+\left(\|f_v^{(s)}\|^2-1\right)\cdot\left(\frac{1+s^2(1-\delta)^2}{\rho(s)}+(\lambda_1+\lambda_2)\right)
\end{aligned}$$

When $s=\pm1$, the above quantity is equal to

$$c(-\lambda_1\lambda_2)\rho(s)+\left(\|f_v^{(s)}\|^2-1\right)\cdot\left(\frac{1+(1-\delta)^2}{\rho(s)}+(\lambda_1+\lambda_2)\right)$$

Now, note that

$$\frac{\langle f_v^{(s)},A_Xf_v^{(s)}\rangle}{\langle f_v^{(s)},f_v^{(s)}\rangle}=\frac{c(-\lambda_1\lambda_2)\rho(s)}{\|f_v^{(s)}\|^2}+\left(1-\frac{1}{\|f_v^{(s)}\|^2}\right)\cdot\left(\frac{1+(1-\delta)^2}{\rho(s)}+(\lambda_1+\lambda_2)\right)\tag{5.10}$$

We now compute $\|f_v^{(s)}\|^2$, and we assume s is either $+1$ or -1.

$$\begin{aligned}
\|f_v^{(s)}\|^2&=\sum_{t=0}^{\infty}\rho(s)^{2t}\sum_{u:d(u,v)=t}\prod_{\{i,j\}\in\mathcal{P}_{uv}}(A_X)_{ij}^2\\
&=\sum_{t=0}^{\infty}\rho(s)^{2t}c(c-1)^{t-1}(-\lambda_1\lambda_2)^t && \text{(by Lemma 5.5.3)}\\
&=\frac{c}{c-1}\sum_{t=0}^{\infty}\left(\frac{(1-\delta)^{2t}}{(c-1)^t(-\lambda_1\lambda_2)^t}\right)(c-1)^t(-\lambda_1\lambda_2)^t\\
&=\frac{c}{c-1}\sum_{t=0}^{\infty}(1-\delta)^{2t}\\
&=\frac{c}{c-1}\cdot\frac{1}{\delta(2-\delta)}
\end{aligned}$$

Plugging this back in to (5.10) gives

$$(5.10)=\delta(2-\delta)(c-1)(-\lambda_1\lambda_2)\rho(s)+\left(\frac{1+(1-\delta)^2}{\rho(s)}+(\lambda_1+\lambda_2)\right)\cdot\left(1-\frac{(c-1)\delta(2-\delta)}{c}\right)$$

$$= \delta(2-\delta)s(1-\delta)\sqrt{(c-1)(-\lambda_1\lambda_2)} +$$
$$\left((1+(1-\delta)^2)\sqrt{(c-1)(-\lambda_1\lambda_2)}\frac{1}{s(1-\delta)} + (\lambda_1+\lambda_2)\right)\cdot\left(1-\frac{(c-1)\delta(2-\delta)}{c}\right)$$

For any $\varepsilon > 0$, we can choose δ small enough so that the above quantity is at least

$$\lambda_1 + \lambda_2 + 2\sqrt{(c-1)(-\lambda_1\lambda_2)} - \varepsilon$$

when $s = 1$ and at most

$$\lambda_1 + \lambda_2 - 2\sqrt{(c-1)(-\lambda_1\lambda_2)} + \varepsilon$$

when $s = -1$.

□

5.5.3 SDP solution for random additive lifts

For $\varepsilon > 0$, consider $f_v^{(1)}$ constructed in the proof of Lemma 5.5.7, for which

$$\langle f_v^{(1)}, A_X f_v^{(1)}\rangle \geqslant (\lambda_1 + \lambda_2 + r_X - \varepsilon)\|f_v^{(1)}\|^2$$

Let L_ε be an integer chosen such that the total ℓ_2 mass of $\frac{f_v^{(1)}}{\|f_v^{(1)}\|}$ on vertices at distance greater than L from v is at most ε. Define g_v as the vector obtained by zeroing out $\frac{f_v^{(1)}}{\|f_v^{(1)}\|}$ on vertices outside $B(v, L)$ and normalizing to make its norm 1, where $B(v, L)$ is the collection of vertices within distance L of v.

For any $\varepsilon' > 0$, we can choose ε so that

$$\langle g_v, A_X g_v\rangle \geqslant \lambda_1 + \lambda_2 + r_X - \varepsilon' \tag{5.11}$$

g_v enjoys the property of being determined by a constant number of vertices, $L_{\varepsilon'}$. For any instance graph G such that there is a unique shortest nomadic walk between any pair of vertices u and v, we can explicitly define

$$g_v(u) = \begin{cases} 0 & \text{if } d(u,v) > L_{\varepsilon'} \\ C \prod\limits_{\{i,j\}\in\mathcal{P}_{uv}} \frac{(1-\delta)(A_X)_{ij}}{\sqrt{(c-1)(-\lambda_1\lambda_2)}} & \mathcal{P}_{uv} \text{ unique shortest nomadic walk from } u \text{ to } v \end{cases}$$

where C is a constant chosen so that g_v has unit norm.

Recall that $\boldsymbol{\mathcal{I}}_n$ is a random signed additive n-lift obtained from a sequence of atoms $\mathcal{A}$.

Definition 5.5.8. Let G be a graph and let $\phi : E(G) \to \{\pm 1\}$ be a signing of the edges. We call a signing ϕ *balanced* if for any cycle given by sequence of edges $e_1, \ldots, e_k$ in $E(H)$, we have $\phi(e_1)\cdots\phi(e_k) = 1$.

We use $A_{\phi(G)}$ to denote the adjacency operator of G signed with respect to ϕ — i.e. $(A_{\phi(G)})_{uv} = \phi(\{u, v\})$ if $\{u, v\}$ is an edge and 0 otherwise.

Lemma 5.5.9. *Suppose ϕ is a balanced signing of G. Then there exists a diagonal sign operator D such that $A_{\phi(G)} = DA_GD^\dagger$.*

Proof. Without loss of generality, assume G is connected. Take a spanning tree of G and root it at some arbitrary vertex r. Let $D_{rr} = 1$ and for P_x a path from r to x let $D_{xx} = \prod_{e \in P_x} \phi(e)$.

It remains to verify that $DA_GD^\dagger = A_{\phi(G)}$. Let P be the path between x and y in the spanning tree. By virtue of ϕ being balanced, we have $\phi(\{x,y\}) \prod_{e\in P} \phi(e) = 1$, which means $\phi(\{x,y\}) = \prod_{e\in P} \phi(e)$. Also, note that $\prod_{e\in P} \phi(e)$ is equal to $\prod_{e\in P_x} \phi(e) \prod_{e\in P_y} \phi(e)$, which is equal to $D_{xx}D_{yy}$. Thus,

$$(A_{\phi(G)})_{ij} = \phi(\{i,j\})(A_G)_{ij} = D_{ii}D_{jj}(A_G)_{ij} = \left(DA_GD^\dagger\right)_{ij}$$

which proves the claim. □

Lemma 5.5.10. *Let X_D be the graph with the adjacency operator $DA_XD^\dagger$ where D is a diagonal sign matrix. There exists D such that X_D covers $\boldsymbol{\mathcal{I}}_n$.*

Proof. When $\boldsymbol{\mathcal{I}}_n$ is generated, (i) the sequence of atoms $\mathcal{A}$ first undergoes an additive n-lift, and then, (ii) the atoms in the lifted graph are given a random balanced signing. The intermediate graph $\widetilde{\boldsymbol{\mathcal{I}}}_n$ between (i) and (ii) is covered by X via a map $\pi : V(X) \to V(\widetilde{\boldsymbol{\mathcal{I}}}_n)$. Once (ii) is performed, construct X' by taking X and setting the signs on all edges in $\pi^{-1}(e)$ to the sign on e for each $e \in E(\boldsymbol{\mathcal{I}}_n)$. X' can be seen as a balanced signing applied on X, and hence there exists such a D by Lemma 5.5.9. □

Definition 5.5.11. Let π be a covering map from appropriate X_D to $\boldsymbol{\mathcal{I}}_n$. Call a vertex $v \in V(\boldsymbol{\mathcal{I}}_n)$ *L-bad* if $B(v,L)$ is not isomorphic to $B(v^*,L)$ where $v^* \in V(X_D)$ is such that $\pi(v^*) = v$.

Remark 5.5.12. The condition of a vertex v in $V(\boldsymbol{\mathcal{I}}_n)$ being L-bad according to Definition 5.5.11 is equivalent to the corresponding variable v' in the constraint graph having a cycle in its distance $2L$-neighborhood.

With the observation of Remark 5.5.12 in hand, we can extract the following as a consequence of [DMO$^+$19b].

Lemma 5.5.13. *The number of K-bad vertices in graph $\boldsymbol{\mathcal{I}}_n$ for constant K is bounded by $O(\log n)$ with probability $1 - o_n(1)$.*

Construct a vector $\widetilde{g}_v$ for each vertex v of $\boldsymbol{\mathcal{I}}_n$.

$$\widetilde{g}_v = \begin{cases} e_v & \text{if } v \text{ is } L_{\varepsilon'}\text{-bad} \\ g_v & \text{otherwise} \end{cases}$$

We are finally ready to prove Theorem 5.5.2.

Proof of Theorem 5.5.2. Let

$$M_+ := \sum_{v \in V(\boldsymbol{\mathcal{I}}_n)} \widetilde{g}_v \widetilde{g}_v^\dagger$$

Writing out $(M_+)_{uu}$ for arbitrary u

$$\begin{aligned}(M_+)_{uu} &= \sum_{v \in V(\boldsymbol{\mathcal{I}}_n)} \widetilde{g}_v(u)\widetilde{g}_v(u) \\ &= \sum_{v \in V(\boldsymbol{\mathcal{I}}_n)} \widetilde{g}_u(v)^2\end{aligned}$$

$$= \|\widetilde{g}_u\|^2 = 1$$

and writing out $\langle A_{\boldsymbol{\mathcal{I}}_n}, M_+\rangle$ gives the following with probability $1 - o_n(1)$.

$$\begin{aligned}
\langle A_{\boldsymbol{\mathcal{I}}_n}, M_+\rangle &= \sum_{v \in V(\boldsymbol{\mathcal{I}}_n)} \langle \widetilde{g}_v, A_{\boldsymbol{\mathcal{I}}_n} \widetilde{g}_v\rangle \\
&= \sum_{\substack{v \in V(\boldsymbol{\mathcal{I}}_n) \\ v \text{ is not } (L_\varepsilon+1)\text{-bad}}} \langle \widetilde{g}_v, A_{\boldsymbol{\mathcal{I}}_n} \widetilde{g}_v\rangle + \sum_{\substack{v \in V(\boldsymbol{\mathcal{I}}_n) \\ v \text{ is } (L_\varepsilon+1)\text{-bad}}} \langle \widetilde{g}_v, A_{\boldsymbol{\mathcal{I}}_n} \widetilde{g}_v\rangle \\
&\geqslant \sum_{\substack{v \in V(\boldsymbol{\mathcal{I}}_n) \\ v \text{ is not } (L_\varepsilon+1)\text{-bad}}} \lambda_1 + \lambda_2 + r_X - \varepsilon' + \sum_{\substack{v \in V(\boldsymbol{\mathcal{I}}_n) \\ v \text{ is } (L_\varepsilon+1)\text{-bad}}} c(\lambda_1\lambda_2) \qquad \text{(by (5.11))} \\
&\geqslant (n - O(\log n))(\lambda_1 + \lambda_2 + r_X - \varepsilon') - O(\log n) \qquad \text{(by Lemma 5.5.13)} \\
&= (1 - o_n(1))(\lambda_1 + \lambda_2 + r_X - \varepsilon')n
\end{aligned}$$

The desired inequality on $\langle A_{\boldsymbol{\mathcal{I}}_n}, M_+\rangle$ can be obtained by choosing ε' small enough and n large enough. The inequality on $\langle A_{\boldsymbol{\mathcal{I}}_n}, M_-\rangle$ can be proved by repeating the whole section and proof by constructing vectors $\widetilde{g}_v$ from $f_v^{(-1)}$. □

5.6 Friedman/Bordenave for additive lifts

Theorem 5.6.1. *Let $\mathcal{A} = (A_1, \ldots, A_c)$ be a sequence of r-vertex atoms with edges weights ± 1. Let $|\mathcal{I}_1|$ denote the instance graph $\mathcal{A}(K_{r,c})$ associated to the base constraint graph when the edge-signs are deleted (i.e., converted to $+1$), and let $|B_1|$ denote the associated nomadic walk matrix. Also, let $\boldsymbol{\mathcal{H}}_n$ denote a random n-lifted constraint graph and $\boldsymbol{\mathcal{I}}_n = \mathcal{A}(\boldsymbol{\mathcal{H}}_n)$ an associated instance graph with 1-wise uniform negations $(\boldsymbol{\xi}^f_{ii'})$. Finally, let $\boldsymbol{B}_n$ denote the nomadic walk matrix for $\boldsymbol{\mathcal{I}}_n$. Then for every constant $\varepsilon > 0$,*

$$\mathbf{Pr}[\rho(\boldsymbol{B}_n) \geqslant \sqrt{\rho(|B_1|)} + \varepsilon] \leqslant \delta,$$

where $\delta = \delta(n)$ is $o_{n\to\infty}(1)$.

Remark 5.6.2. It might seem that our bound involving $|B_1|$ may be poor, given that it ignores sign information from the atoms. However, it is in fact sharp, and the reason is that the main contribution to $\rho(\boldsymbol{B}_n)$ when using the Trace Method is from walks in which almost all edges are traversed twice. And if an edge is traversed twice, it of course does not matter if its sign is -1 or $+1$.

Remark 5.6.3. In fact, it is evident from the theorem statement that without loss of generality we may assume that the atoms are unweighted — i.e., that all weights are $+1$. The reason is that for each constraint f in group j, if we multiply $\boldsymbol{\xi}^f_{ii'}$ by the fixed value $A_j[i, i']$, the resulting signs remain 1-wise uniform — and this has the effect of eliminating all signs from the atoms. Thus henceforth we will indeed assume that the original atoms are all unweighted.

The idea of Friedman/Bordenave proofs. The standard method for trying to prove a theorem such as Theorem 5.6.1 involves applying the Trace Method to $\boldsymbol{B}_n$. Since $\boldsymbol{B}_n$ is not a self-adjoint operator, a natural way to do this is to consider $\mathrm{tr}(\boldsymbol{B}_n^\ell \boldsymbol{B}_n^{*\ell})$ for some large ℓ. Roughly speaking, this counts the number of closed walks that walk nomadically in $\boldsymbol{\mathcal{I}}_n$ for the first ℓ steps, and then walk nomadically in the *reverse* of $\boldsymbol{\mathcal{I}}_n$ for the next ℓ steps. A major difficulty is the following: the Trace Method naturally incurs an "extra" factor of n, and to overcome this one wants to choose $\ell \gg \log n$. However, $\Theta(\log n)$ is precisely the radius at which random constraint graphs become dramatically non-tree-like; i.e., they are likely to encounter nontrivial cycles. Based on Friedman's work, Bordenave overcomes this difficulty as follows: First, ℓ is set to $c \log n$ for some small positive constant $c > 0$. Nomadic walks of this length may well encounter cycles, but one can show that with high probability, they will not encounter *tangles* — meaning, *more than one* cycle in a radius of ℓ. (This crucial concept of "tangles" was isolated by Friedman and refined by Bordenave.) Now we set $k = \omega_n(1)$ to be a slowly growing quantity and consider length-$2k\ell$ walks formed by doing ℓ nomadic steps, then ℓ nomadic reverse-steps, all k times in succession. In other words, we consider $\mathrm{tr}((\boldsymbol{B}_n^\ell \boldsymbol{B}_n^{*\ell})^k)$. On one hand, since $2k\ell \gg \log n$, bounding this quantity will be sufficient to overcome the n-factor inherent in the Trace Method. On the other hand, using tangle-freeness at radius ℓ along with very careful combinatorial counting allows us to bound the number of closed length-$2k\ell$ walks.

Our proof follows this methodology and draws ideas from Bordenave's original proof from [Bor15] as well as [DMO$^+$19b] and [BDH18]. However, our main technical lemma, Lemma 5.6.24, uses a new tool that takes advantage of the random negations our model employs that simplifies the equivalent proofs in the three mentioned papers and also allows us to generalize it to our model.

5.6.1 Trace Method setup, and getting rid of tangles

To begin carrying out this proof strategy, we first define tangle-freeness.

Definition 5.6.4 (Tangles-free)**.** Let G be an undirected graph. A vertex v is said to be *ℓ-tangle-free within G* if the subgraph of G induced by v's distance-4ℓ neighborhood contains at most one cycle.[11]

It is straightforward to show that random lifts have all vertices $\Theta(\log n)$-tangle-free; we can quote the relevant result directly from Bordenave (Lemma 27 from [Bor15]):

Proposition 5.6.5. *There is a universal constant $\kappa > 0$ depending only on r, c such that, for $\ell = \kappa \log n$, a random n-lift $\boldsymbol{\mathcal{H}}$ of $K_{r,c}$ has all vertices ℓ-tangle free, except with probability $O(1/n^{.99})$.*

We now begin the application of the Trace Method. We have:

$$\mathrm{tr}((\boldsymbol{B}_n^\ell \boldsymbol{B}_n^{*\ell})^k) = \sum_{\vec{e}_0,\dots,\vec{e}_{2k\ell-1},\vec{e}_{2k\ell}=\vec{e}_0} \boldsymbol{B}_n[\vec{e}_0,\vec{e}_1]\cdots\boldsymbol{B}_n[\vec{e}_{\ell-1},\vec{e}_\ell]\boldsymbol{B}_n^*[\vec{e}_\ell,\vec{e}_{\ell+1}]\cdots\boldsymbol{B}_n^*[\vec{e}_{2\ell-1},\vec{e}_{2\ell}]\cdots\boldsymbol{B}_n^*[\vec{e}_{2k\ell-1},\vec{e}_{2k\ell}]$$

$$= \sum_{\vec{e}_0,\dots,\vec{e}_{2k\ell-1},\vec{e}_{2k\ell}=\vec{e}_0} \boldsymbol{B}_n[\vec{e}_0,\vec{e}_1]\cdots\boldsymbol{B}_n[\vec{e}_{\ell-1},\vec{e}_\ell]\boldsymbol{B}_n[\vec{e}_{\ell+1},\vec{e}_\ell]\cdots\boldsymbol{B}_n[\vec{e}_{2\ell},\vec{e}_{2\ell-1}]\cdots\boldsymbol{B}_n[\vec{e}_{2k\ell},\vec{e}_{2k\ell-1}]$$

[11] We chose the factor 4 here for "safety". For quantitative aspects of our theorem, constant factors on ℓ will be essentially costless.

$$= \sum \mathrm{wt}(e_1) N_{\vec{e}_0,\vec{e}_1} \cdots \mathrm{wt}(e_\ell) N_{\vec{e}_{\ell-1},\vec{e}_\ell} \mathrm{wt}(e_\ell) N_{\vec{e}_\ell^{-1},\vec{e}_{\ell+1}^{-1}} \cdots \mathrm{wt}(e_{2\ell-1}) N_{\vec{e}_{2\ell-1}^{-1},\vec{e}_{2\ell}^{-1}} \cdots \mathrm{wt}(e_{2k\ell-1}) N_{\vec{e}_{2k\ell-1}^{-1},\vec{e}_{2k\ell}^{-1}}, \tag{5.12}$$

where $\mathrm{wt}(e)$ is the sign on edge e coming from the random 1-wise negations (it is the same for both directed versions of the edge), and where $N_{\vec{e},\vec{f}}$ is an indicator that $(\vec{e},\vec{f})$ forms a length-2 nomadic walk. Roughly speaking, this quantity counts (with some ± 1 sign) closed walks in $\boldsymbol{\mathcal{I}}_n$ consisting of $2k$ consecutive nomadic walks of length ℓ. However, there is some funny business concerning the joints between these nomadic walks. To be more precise, in each of the $2k$ segments we have a nomadic walk of $\ell+1$ edges; and, the last edge in each segment must be the reverse of the first edge in the subsequent segment. We will call these necessarily-duplicated edges "spurs". Furthermore, when computing the sign with which the closed walk is counted, spurs' signs are counted either zero times or twice, depending on the parity of the segment. Hence they are effectively discounted, since $(-1)^2 = (-1)^0 = +1$. Let us make some definitions encapsulating all of this.

Definition 5.6.6 (Nomadic linkages, and spurs)**.** In an instance graph, a $(2k \times \ell)$-*nomadic linkage* $\mathcal{L}$ is the concatenation of $2k$ many nomadic walks ("segments"), each of length $\ell+1$, in which the last directed edge of each walk is the reverse of first directed edge of the subsequent walk (including wrapping around from the $2k$th segment to the 1st). These $2k$ directed edges which are necessarily the reverse of the preceding directed edge are termed *spurs*. The *weight* of $\mathcal{L}$, denoted $\mathrm{wt}(\mathcal{L})$, is the product of the signs of the non-spur edges in $\mathcal{L}$.

Definition 5.6.7 (Nonbacktracking $\mathcal{A}$-linkages)**.** Recall that, strictly speaking, the nomadic property requires "remembering" which atom each edge comes from. Thus the $\mathcal{L}$ above is really associated to what we will call a $(2k \times 2\ell)$-*nonbacktracking $\mathcal{A}$-linkage* — call it $\mathcal{C}$ — in the underlying constraint graph. Formally:

- ("linkage") $\mathcal{C}$ is a closed concatenation of $2k$ walks (called "segments") in the constraint graph, each consisting of $\ell+1$ length-2 variable-constraint-variable subpaths. The last such length-2 subpath in each segment ("spur") is equal to (the reverse of) the first length-2 subpath in the subsequent segment (including wraparound from the $2k$th segment to the 1st).
- ("$\mathcal{A}$-linkage") For each length-2 subpath (v, f, v') in $\mathcal{C}$, where v is in variable group i, f is in constraint group j, and v' is in variable group i', it holds that $\{i, i'\}$ is an edge in A_j.
- ("nonbacktracking") Each of the $2k$ segments is a nonbacktracking walk of length $2(\ell+1)$ in the constraint graph.

We write $\mathrm{wt}(\mathcal{C}) \in \{\pm 1\}$ for the weight of the associated nomadic linkage in the instance graph. Given these definitions, (5.12) tells us:

$$\mathrm{tr}((\boldsymbol{B}_n^\ell \boldsymbol{B}_n^{*\ell})^k) = \sum_{\substack{(2k\times 2\ell)\text{-nonbacktracking} \\ \mathcal{A}\text{-linkages } \mathcal{C} \text{ in } \boldsymbol{\mathcal{H}}_n}} \mathrm{wt}(\mathcal{C}). \tag{5.13}$$

Next, we make the observation that *if* $\boldsymbol{\mathcal{H}}_n$ proves to have all vertices ℓ-tangle-free, then we would get the same result if we only summed over "externally tangle-free" linkages.

Definition 5.6.8 (Externally tangle-free linkages)**.** We say that a $(2k \times 2\ell)$-nonbacktracking linkage in a constraint graph $\mathcal{H}_n$ is *externally ℓ-tangle-free* if every vertex it touches is ℓ-tangle-

free within $\mathcal{H}_n$. (The "externally" adjective emphasizes that we are concerned with cycles not just within the linkage's edges, but also among nearby edges of $\mathcal{H}_n$.)

Thus in light of Proposition 5.6.5 we have:

Lemma 5.6.9. *Provided $\ell \leqslant \kappa \log n$ for a certain universal $\kappa > 0$, we get that $\mathrm{tr}((\boldsymbol{B}_n^\ell \boldsymbol{B}_n^{*\ell})^k) = \boldsymbol{S}$ holds except with probability $O(1/n^{.99})$, where*

$$\boldsymbol{S} \coloneqq \sum_{\substack{(2k\times 2\ell)\text{-nonbacktracking} \\ \text{externally } \ell\text{-tangle-free} \\ \mathcal{A}\text{-linkages } \mathcal{C} \text{ in } \boldsymbol{\mathcal{H}}_n}} \mathrm{wt}(\mathcal{C}).$$

In order to apply Markov's inequality later, we will need the following technical claim:

Claim 5.6.10. $\boldsymbol{S}$ is a nonnegative random variable.

Proof. Given $\boldsymbol{\mathcal{I}}_n$, recall that

$$\boldsymbol{B}_n^\ell[\vec{e}, \vec{f}] = \sum_{\substack{\text{nomadic walks} \\ \vec{e}=\vec{e}_0,\vec{e}_1,\ldots,\vec{e}_\ell=\vec{f} \text{ in } \boldsymbol{\mathcal{I}}_n}} \mathrm{wt}(e_1)\mathrm{wt}(e_2)\cdots\mathrm{wt}(e_\ell).$$

Using a key idea of Bordenave (based on the "selective trace" of Friedman), define the related operator $\boldsymbol{B}_n^{(\ell)}$ via

$$\boldsymbol{B}_n^{(\ell)}[\vec{e}, \vec{f}] = \sum_{\substack{\textit{externally } \ell\textit{-tangle-free} \text{ nomadic walks} \\ \vec{e}=\vec{e}_0,\vec{e}_1,\ldots,\vec{e}_\ell=\vec{f} \text{ in } \boldsymbol{\mathcal{I}}_n}} \mathrm{wt}(e_1)\mathrm{wt}(e_2)\cdots\mathrm{wt}(e_\ell),$$

where again the walk is said to be "externally ℓ-tangle-free" if every vertex it touches is ℓ-tangle-free with $\boldsymbol{\mathcal{H}}_n$. Then very similar to the analysis that gave us (5.12) and (5.13), we get that

$$\boldsymbol{S} = \mathrm{tr}((\boldsymbol{B}_n^{(\ell)}(\boldsymbol{B}_n^{(\ell)})^*)^k).$$

Thus $\boldsymbol{S}$ is visibly always nonnegative, being the trace of the kth power of the positive semidefinite matrix $\boldsymbol{B}_n^{(\ell)}(\boldsymbol{B}_n^{(\ell)})^*$. □

With these results in place, we can proceed to the main goal of the Trace Method: bounding $\mathbf{E}[\boldsymbol{S}]$. Such a bound can be used in the following lemma:

Lemma 5.6.11. *Assume that $\ell \leqslant \kappa \log n$ and $k\ell = \omega(\log n)$. Then from $\mathbf{E}[\boldsymbol{S}] \leqslant R$ we may conclude that $\rho(\boldsymbol{B}_n) \leqslant (1 + o_n(1)) \cdot R^{\frac{1}{2k\ell}}$ holds, except with probability $O(1/n^{.99})$.*

Proof. Let $\boldsymbol{T} = \mathrm{tr}((\boldsymbol{B}_n^\ell \boldsymbol{B}_n^{*\ell})^k)$. On one hand, with λ denoting eigenvalues and σ denoting singular values, we have

$$\boldsymbol{T} \geqslant \lambda_{\max}((\boldsymbol{B}_n^\ell \boldsymbol{B}_n^{*\ell})^k) = \lambda_{\max}\left(\sqrt{\boldsymbol{B}_n^\ell \boldsymbol{B}_n^{*\ell}}\right)^{2k} = \sigma_{\max}(\boldsymbol{B}_n^\ell)^{2k} \geqslant \rho(\boldsymbol{B}_n^\ell)^{2k} = \rho(\boldsymbol{B}_n)^{2k\ell}.$$

On the other hand, since $\boldsymbol{S}$ is a nonnegative random variable (Claim 5.6.10), we can apply Markov's Inequality to deduce that $\boldsymbol{S} \leqslant n \cdot R$ except with probability at most $1/n$. Now from Lemma 5.6.9 we may infer that except with probability $O(1/n^{.99})$,

$$\boldsymbol{T} = \boldsymbol{S} \leqslant n \cdot R \quad \Longrightarrow \quad \rho(\boldsymbol{B}_n)^{2k\ell} \leqslant n \cdot R.$$

The result now follows by taking $2k\ell$-th roots. □

5.6.2 Eliminating singletons, and reduction to counting

Our next step toward bounding $\mathbf{E}[\boldsymbol{S}]$ is typical of the Trace Method: Rather than first choosing $\boldsymbol{\mathcal{H}}_n$ randomly and then summing over the linkages therein, we instead sum over all *potentially-appearing* linkages and insert an indicator that they actually appear in the realized random constraint graph. Defining

$$\mathcal{K}_n = \text{the "complete" constraint graph with } cn \text{ constraint vertices and } rn \text{ variable vertices,}$$

this means that

$$\boldsymbol{S} = \sum_{\substack{(2k\times 2\ell)\text{-nonbacktracking} \\ \mathcal{A}\text{-linkages } \mathcal{C} \text{ in } \mathcal{K}_n}} 1[\mathcal{C} \text{ is in } \boldsymbol{\mathcal{H}}_n] \cdot 1[\mathcal{C} \text{ is externally } \ell\text{-tangle-free within } \boldsymbol{\mathcal{H}}_n] \cdot \mathrm{wt}_{\boldsymbol{\mathcal{I}}_n}(\mathcal{C}). \tag{5.14}$$

Here we wrote $\mathrm{wt}_{\boldsymbol{\mathcal{I}}_n}(\mathcal{C})$ to emphasize that even once $\mathcal{C}$ is in $\boldsymbol{\mathcal{H}}_n$ and is externally ℓ-tangle-free, its weight is still a random variable arising from the 1-wise uniform negations. These negations will create another simplification (one not available to Friedman/Bordenave). For this we will need another definition:

Definition 5.6.12 (Singleton-free $\mathcal{C}$'s)**.** Let $\mathcal{C}$ be a $(2k \times 2\ell)$-nonbacktracking circuit in $\mathcal{K}_n$. If there is an atom vertex that is passed through exactly once, we call it a *singleton*. If $\mathcal{C}$ contains no singleton, we call it *singleton-free*.

Referring to (5.14), consider $\mathbf{E}[\boldsymbol{S}]$. If $\mathcal{C}$ contains any singleton, then it will contribute 0 to this expectation. The reason is that, provided $\mathcal{C}$ appears in $\boldsymbol{\mathcal{H}}_n$ and is externally ℓ-tangle-free therein, the 1-wise uniform negations will assign a uniformly random ± 1 sign to the edge engendered by $\mathcal{C}$'s singleton, and this sign will be independent of all other signs that go into $\mathrm{wt}_{\boldsymbol{\mathcal{I}}_n}(\mathcal{C})$. On the other hand, when $\mathcal{C}$ is singleton-free, we will simply upper-bound the (conditional) expectation of $\mathrm{wt}_{\boldsymbol{\mathcal{I}}_n}(\mathcal{C})$ by $+1$. We conclude that

$$\mathbf{E}[\boldsymbol{S}] \leqslant \sum_{\substack{(2k\times 2\ell)\text{-nonbacktracking} \\ \textit{singleton-free} \\ \mathcal{A}\text{-linkages } \mathcal{C} \text{ in } \mathcal{K}_n}} \mathbf{Pr}[\mathcal{C} \text{ is in } \boldsymbol{\mathcal{H}}_n \text{ and is externally } \ell\text{-tangle-free therein}]. \tag{5.15}$$

Let us now begin to simplify the probability calculation.

Definition 5.6.13 ($E(\mathcal{C})$, $V(\mathcal{C})$, $G(\mathcal{C})$)**.** Let $\mathcal{C}$ be a $(2k \times 2\ell)$-nonbacktracking $\mathcal{A}$-linkage in $\mathcal{K}_n$. Write $E(\mathcal{C})$ for the set of undirected edges in $\mathcal{K}_n$ formed by "undirecting" all the directed edges in $\mathcal{C}$ (this includes reducing from a multiset to a set, if necessary). Then let $G(\mathcal{C})$ denote the undirected subgraph of $\mathcal{K}_n$ induced by $E(\mathcal{C})$, and write $V(\mathcal{C})$ for its vertices.

Let's simplify the "tangle-freeness" situation.

Definition 5.6.14 (Internal tangle-free linkages)**.** We say that a $(2k \times 2\ell)$-nonbacktracking linkage $\mathcal{C}$ in $\mathcal{K}_n$ is *internally ℓ-tangle-free* if every vertex it touches is ℓ-tangle-free *within* $G(\mathcal{C})$.

We certainly have:

$$\text{linkage } \mathcal{C} \text{ not even internally } \ell\text{-tangle-free} \implies \mathbf{Pr}[\mathcal{C} \text{ is in } \boldsymbol{\mathcal{H}}_n \text{ and is externally } \ell\text{-tangle-free therein}] = 0.$$

Thus we can restrict the sum in (5.15) to internally ℓ-tangle-free linkages. Having done that, we will upper bound the sum by dropping this insistence on *external* tangle-freeness. Thus

$$\mathbf{E}[\boldsymbol{S}] \leqslant \sum_{\substack{(2k\times 2\ell)\text{-nonbacktracking} \\ \textit{interally } \ell\textit{-tangle-free}\text{, singleton-free} \\ \mathcal{A}\text{-linkages } \mathcal{C} \text{ in } \mathcal{K}_n}} \mathbf{Pr}[\mathcal{C} \text{ is in } \boldsymbol{\mathcal{H}}_n]. \tag{5.16}$$

We will now bound $\mathbf{Pr}[\mathcal{C} \text{ is in } \boldsymbol{\mathcal{H}}_n]$, so as to reduce all our remaining problems to counting. Towards this, recall that $\boldsymbol{\mathcal{H}}_n$ is a random n-lift of the complete graph $K_{r,c}$. One thing this implies is that every group-i variable-vertex in $\boldsymbol{\mathcal{H}}_n$ will have exactly one edge to each of c groups of constraint-vertices, and vice versa. Let us codify the $\mathcal{C}$'s that don't flagrantly violate this property:

Definition 5.6.15 (Valid $\mathcal{C}$'s)**.** We say a $(2k \times 2\ell)$-nonbacktracking $\mathcal{A}$-linkage $\mathcal{C}$ in $\mathcal{K}_n$ is *valid* if $G(\mathcal{C})$ has the property that every variable-vertex in it is connected to at most 1 constraint-vertex from each of the c groups, and each constraint-vertex is connected to at most 1 variable-vertex from each of the r groups.

Evidently, $\mathbf{Pr}[\mathcal{C} \text{ is in } \boldsymbol{\mathcal{H}}_n] = 0$ if $\mathcal{C}$ is invalid. Thus from (5.16) we can deduce:

$$\mathbf{E}[\boldsymbol{S}] \leqslant \sum_{\substack{(2k\times 2\ell)\text{-nonbacktracking} \\ \textit{valid}\text{, internally } \ell\text{-tangle-free, singleton-free} \\ \mathcal{A}\text{-linkages } \mathcal{C} \text{ in } \mathcal{K}_n}} \mathbf{Pr}[\mathcal{C} \text{ is in } \boldsymbol{\mathcal{H}}_n]. \tag{5.17}$$

Next, it is straightforward to show the following lemma (see Proposition A.8 of [DMO$^+$19b] for essentially the same observation):

Lemma 5.6.16. *If $\mathcal{C}$ is a valid $(2k \times 2\ell)$-nonbacktracking $\mathcal{A}$-linkage in $\mathcal{K}_n$, and $k\ell = o(\sqrt{n})$, then*

$$\mathbf{Pr}[\mathcal{C} \text{ is in } \boldsymbol{\mathcal{H}}_n] = (1 + o_n(1)) \cdot n^{-|E(\mathcal{C})|}.$$

Proof. (Sketch.) Proceed through the edges in $E(\mathcal{C})$ in an arbitrary order. Each has approximately a $1/n$ chance of appearing in $\boldsymbol{\mathcal{H}}_n$, even conditioned on the appearance of the preceding edges. For example, this is exactly true for the first edge. For subsequent edges $e = \{u, v\}$, validity ensures that no preceding edge already connects u to a vertex in v's part, or vice versa. Thus the conditional probability of e appearing in $\boldsymbol{\mathcal{H}}_n$ is essentially the probability that a particular edge appears in a random matching on $n+n$ vertices (which is $1/n$), except that a "small" number of vertex pairs may already have been matched. This "small" quantity is at most $|E(\mathcal{C})| \leqslant 4k\ell$, so the $1/n$ probability becomes $1/(n - 4k\ell)$ at worst. Multiplying these conditional probabilities across all $|E(\mathcal{C})|$ edges yields a quantity that is off from $n^{-|E(\mathcal{C})|}$ by a factor of at most $(1 + O(k\ell)/n)^{4k\ell} \leqslant 1 + o_n(1)$, the inequality using $(k\ell)^2 = o(n)$. □

Combining this lemma with (5.17) and Lemma 5.6.11, we are able to reduce bounding $\rho(\boldsymbol{B}_n)$ to a counting problem:

Lemma 5.6.17. *Assume that $\ell \leqslant \kappa \log n$ and $\omega(\log n) < k\ell < o(\sqrt{n})$. Then except with probability $O(1/n^{.99})$,*

$$\rho(\boldsymbol{B}_n) \leqslant (1 + o_n(1)) \cdot R^{\frac{1}{2k\ell}}, \quad \textit{where } R \coloneqq \sum_{\substack{(2k\times 2\ell)\text{-nonbacktracking} \\ \text{valid, internally } \ell\text{-tangle-free, singleton-free} \\ \mathcal{A}\text{-linkages } \mathcal{C} \text{ in } \mathcal{K}_n}} n^{-|E(\mathcal{C})|}.$$

5.6.3 Tangle-free, singleton-free linkages are nearly duplicative

Our goal in this subsection is to show that each linkage $\mathcal{C}$ we sum over in Lemma 5.6.17 is "nearly duplicative": the number of variable-vertices is at most $(1 + o(1))k\ell$, and the same is true of constraint-vertices — even though the obvious a priori upper bound for each of them is $2k\ell$. This factor-$\frac{1}{2}$ savings is precisely the source of the square-root in Theorem 5.6.1. We begin with a graph-theoretic lemma and then deduce the nearly-duplicative property.

Lemma 5.6.18. *Let $\mathcal{C}$ be a $(2k \times 2\ell)$-nonbacktracking, internally ℓ-tangle-free linkage in $\mathcal{K}_n$. Assume $\log(k\ell) = o(\ell)$. Then $G(\mathcal{C})$ has at most $O(k \log(k\ell))$ vertices of degree exceeding 2.*

Proof. For brevity, let us write $G = G(\mathcal{C})$, $w = |V(\mathcal{C})|$, and note that we have a trivial upper bound of $w \leqslant 4k\ell$. Let t denote the number of cycles of length at most ℓ in G. By deleting at most t edges, we can form a graph $\widetilde{G}$ with girth at least ℓ. A theorem of Alon, Hoory, and Linial [AHL02] implies that any (possibly irregular) graph with w vertices and girth at least ℓ must have average degree at most $2 + O(\log(w)/\ell)$ (this uses $\log(w) = o(\ell)$). Thus $\widetilde{G}$ has such a bound on its average degree. After restoring the deleted edges, we can still conclude that the average degree in G is at most $2 + O(\log(w)/\ell) + \frac{2t}{w}$. Writing w_1, w_2, w_{3^+} for the number of vertices in G of degree 1, 2, and 3-or-more respectively, this means

$$2 + O(\log(w)/\ell) + \frac{2t}{w} \geqslant \frac{w_1 + 2w_2 + 3w_{3^+}}{w} = \frac{w_1 + 2(w - w_1 - w_{3^+}) + 3w_{3^+}}{w} = 2 - \frac{w_1}{w} + \frac{w_{3^+}}{w}$$
$$\implies \quad w_{3^+} \leqslant O(w \log(w)/\ell) + w_1 + 2t.$$

The first term here is $O(k \log(k\ell))$ as desired, since $w \leqslant 4k\ell$. We will also show the next two terms are $O(k)$. Regarding w_1, degree-1 vertices in G can only arise from the spurs of $\mathcal{C}$, and hence $w_1 \leqslant 2k$. Finally, $2t \leqslant O(k)$ follows from the below claim combined with $w \leqslant 4k\ell$:

$$t \leqslant \frac{w}{2\ell} + 1. \tag{5.18}$$

We establish (5.18) using the tangle-free property of $\mathcal{C}$. Recall that t is the number of "short" cycles in G, meaning cycles of length at most ℓ. By the ℓ-tangle-free property of $\mathcal{C}$ (recalling the factor 4 in its definition), every $v \in V$ has at most one short cycle within distance 3ℓ of it. Thus if we choose paths in G that connect all short cycles (recall G is connected), then to each short cycle we can uniquely charge at least $3\ell - 1 \geqslant 2\ell$ vertices from these paths. It follows that $w = |V| \geqslant 2\ell(t - 1)$, establishing (5.18). □

Corollary 5.6.19. *In the setting of Lemma 5.6.18, assume also that $\mathcal{C}$ is singleton-free and valid. Then the number of variable-vertices $\mathcal{C}$ visits is at most $k\ell + O(k \log(k\ell))$, and the same is true of constraint-vertices.*

Proof. Think of $\mathcal{C}$ as a succession of $2k(\ell + 1)$ "two-steps", where a two-step is a length-2 directed path going from a variable-vertex, to a constraint-vertex, to a (distinct) variable-vertex. Call two such two-steps "duplicates" if they use the same three variables (possibly going in the opposite direction). We claim that "almost all" two-steps have at least one duplicate. To see this, consider the constraint-vertex in some two-step a. Since $\mathcal{C}$ is singleton-free, at least one other two-step b must pass through the constraint-vertex of a. If b is not a duplicate of a, then

this constraint-vertex will have degree exceeding 2 in $G(\mathcal{C})$. By Lemma 5.6.18 there are at most $O(k\log(k\ell))$ such constraint-vertices. Further, by validity each constraint-vertex can support at most $\binom{r}{2} = O(1)$ unduplicated two-steps. Thus at most $O(k\log(k\ell))$ of the $2k(\ell+1)$ two-steps are unduplicated.

Now imagine we walk through the two-steps of $\mathcal{C}$ in succession. Each two-step can visit at most one "new" variable-vertex and one "new" constraint-vertex. However each two-step which is a duplicate of a previously-performed two-step visits no new vertices. Among the $2k(\ell+1)$ two-steps, at most $O(k\log(k\ell))$ are unduplicated. Thus at least $(2k(\ell+1) - O(k\log(k\ell)))/2 = k(\ell+1) - O(k\log(k\ell))$ two-steps are duplicates of previously-performed two-steps. It follows that at most $k(\ell+1)+O(k\log(k\ell))$ two-steps visit any new vertex. This completes the proof. □

5.6.4 The final countdown

We now wish to count the objects summed in the definition of R from Lemma 5.6.17. The remainder of this section will be devoted to proving:

Theorem 5.6.20. *For every $\varepsilon > 0$, except with probability $O(1/n^{.99})$,*

$$\rho(\boldsymbol{B}_n) \leqslant (1+o_n(1))\cdot(1+\varepsilon)\cdot\sqrt{\rho(|B_1|)}.$$

The bulk of the technical matter in the proof of Theorem 5.6.20 will involve analyzing

$$(2k\times 2\ell)\text{-nonbacktracking, valid, internally } \ell\text{-tangle-free, singleton-free, } \mathcal{A}\text{-linkages } \mathcal{C} \quad (5.19)$$

in $\mathcal{K}_n$.

Definition 5.6.21 (Steps: stale, fresh, and boundary)**.** We call each of the $4k(\ell+1)$ directed edges from which $\mathcal{C}$ is composed a *step*. If we imagine traversing these steps in order, they "reveal" vertices and edges of $G(\mathcal{C})$ as we go along. We call a step *stale* if the edge it traverses was previously traversed in $\mathcal{C}$ (in some direction). Note that both endpoints of the edge must also have been previously visited. Otherwise, if the step traverses a "new" edge, it will be designated either "fresh" or "boundary". It is designated *fresh* if the vertex it reaches was never previously visited in $\mathcal{C}$. Otherwise, the step is *boundary*; i.e., the step goes between two previously-visited vertices, but along a new edge. For the purposes of defining fresh/boundary, we specify that the initial vertex of $\mathcal{C}$ is always considered to be "previously visited".

The following facts are immediate:

Fact 5.6.22. *The number of fresh steps in $\mathcal{C}$ is $|V(\mathcal{C})|-1$. (The -1 accounts for the fact that the initial vertex is considered "previously visited".) Since the number of fresh and boundary steps together is $|E(\mathcal{C})|$, it follows that the number of boundary steps is $|E(\mathcal{C})| - |V(\mathcal{C})| + 1$.*

Definition 5.6.23. We write $\mathrm{Lkgs}(f,b)$ for the collection of linkages as in (5.19) having exactly f fresh edges and b boundary edges.

Our goal is to show:

Lemma 5.6.24. *For every $\hat{\rho} > \rho(|B_1|)$ we have:*

$$|\mathrm{Lkgs}(f,b)| \leqslant \mathrm{poly}(k,\ell)^{b+k}\cdot n^{f+1}\cdot\hat{\rho}^{f/2}$$

where the constants in the poly *factor depend on $\hat{\rho}$.*

Before proving this lemma, observe that many linkages are the same modulo the labels between 1 and n that are defined by the lifting. To make this formal we first introduce some notation and follow by using it to aid in the proof of Lemma 5.6.24.

Given a linkage $\mathcal{C}$ we write $\mathcal{C} = ((v_1, i_1), (v_2, i_2), \ldots, (v_{4k(\ell+1)}, i_{4k(\ell+1)}))$, where (v_j, i_j) are vertices from $\mathcal{K}_n$ and v_j indicates the base vertex (from $K_{r,c}$) and i_j is an integer (between 1 and n) that indicates the lifted copy. This notation means that $\mathcal{C}$ traverses this sequence of vertices in this order.

Definition 5.6.25 (Isomorphism of linkages)**.** Given two linkages $\mathcal{C}$ and $\mathcal{C}'$ that visit $|V(\mathcal{C})| = |V(\mathcal{C}')|$ vertices, we say they are *isomorphic* if are the same modulo the labels between 1 and n that are defined by the lifting. Formally, letting $\mathcal{C} = ((v_1, i_1), \ldots, (v_{4k(\ell+1)}, i_{4k(\ell+1)}))$ and $\mathcal{C}' = ((v'_1, i'_1), \ldots, (v'_{4k(\ell+1)}, i'_{4k(\ell+1)}))$, there exist permutations π_v on $[n]$ for each $v \in V(K_{r,c})$ such that for all j we have $v'_j = v_j$ and $i'_j = \pi_{v_j}(i_j)$.

This isomorphism relation induces equivalence classes for which we want to assign representative elements. We do so as follows.

Definition 5.6.26 (Canonical linkages)**.** A linkage $\mathcal{C}$ is said to be *canonical* if for every vertex $v \in K_{r,c}$, if $\mathcal{C}$ visits j distinct lifted copies of v then it first visits $(v, 1)$, then $(v, 2)$, ..., and finally (v, j). We write $\mathrm{Lkgs}^c(f, b)$ for the collection of *canonical* linkages as in (5.19) having exactly f fresh steps and b boundary steps.

Proposition 5.6.27. $|\mathrm{Lkgs}(f, b)| \leqslant n^{f+1} |\mathrm{Lkgs}^c(f, b)|$.

Proof. It suffices to show that for every canonical linkage $\mathcal{C} \in \mathrm{Lkgs}^c(f, b)$, it has at most n^{f+1} isomomorphic linkages $\mathcal{C}' \in \mathrm{Lkgs}(f, b)$. By Fact 5.6.22, $\mathcal{C}$ visits exactly $f + 1$ distinct vertices, call them $\{(v^{(1)}, i^{(1)}), \ldots, (v^{(f+1)}, i^{(f+1)})\}$. Every isomorphic $\mathcal{C}'$ may be obtained by taking a list of numbers $(i'_1, \ldots, i'_{f+1}) \in [n]^{f+1}$ and replacing all appearances of $(v^{(j)}, i^{(j)})$ in $\mathcal{C}$ with $(v^{(j)}, i'_j)$. (Not all such lists lead to isomorphic $\mathcal{C}'$, but we don't mind overcounting.) This completes the proof, as there are n^{f+1} such lists. □

We now have all the tools to prove the desired lemma.

Proof of Lemma 5.6.24. With Proposition 5.6.27 in place, it suffices to bound the number of canonical linkages as follows:

$$|\mathrm{Lkgs}^c(f, b)| \leqslant \mathrm{poly}(k, \ell)^{b+k} \cdot \hat{\rho}^{f/2}.$$

Our strategy is to give an encoding of linkages in $\mathrm{Lkgs}^c(f, b)$, and then bound the number of possible encodings. Let $\mathcal{C}$ be an arbitrary linkage in $\mathrm{Lkgs}^c(f, b)$. To encode $\mathcal{C}$, we first partition it into $2k$ many "$2(\ell+1)$-segments", each of which corresponds to nonbacktracking walks between spurs,and specify how to encode each $2(\ell+1)$-segment. We then partition each $2(\ell+1)$-segment into maximal contiguous blocks of the same type of step ("type" as in Definition 5.6.21) and store an encoding of information about the steps therein. Ultimately, it will be possible to uniquely decipher $\mathcal{C}$ from its constructed encoding.

Towards describing our encoding, we first define the sequence S_{visited}, constructed from the $f + 1$ vertices in $V(\mathcal{C})$ sorted in increasing order of first-visit time.

Encoding positions of blocks. We define P_{fresh}, P_{boundary} and P_{stale}, which are sequences noting the starting positions and ending positions of fresh, boundary, and stale blocks respectively, in the order visited in $\mathcal{C}$.

Encoding fresh steps. Let S_{fresh} be the sequence obtained by replacing each vertex of S_{visited} with its corresponding base vertex in $K_{r,c}$.

Encoding boundary steps. Let β be a block of boundary steps $(v_0, v_1), \ldots, (v_{|\beta|-1}, v_{|\beta|})$. Let t_i be such that v_i is the t_i-th vertex in S_{visited}. We define $\text{Enc}_b(\beta)$ as the sequence $(t_0, t_1), \ldots, (t_{|\beta|-1}, t_{|\beta|})$. Let $\beta_1, \ldots, \beta_T$ be the blocks of boundary steps in the order in which they appear in $\mathcal{C}$. We store the concatenation of $\text{Enc}_b(\beta_1), \ldots, \text{Enc}_b(\beta_T)$, which we call S_{boundary}.

Encoding stale steps. For each block β of stale steps, let u be the first vertex and v be the last vertex of β, and let $p(\beta)$ be the position in $\mathcal{C}$ where the block β starts. Let $\mathcal{S}_{p(\beta),uv,|\beta|}$ denote the list (in, say, lexicographic order) of all possible nonbacktracking walks from u to v of length $|\beta|$ that only use edges visited by $\mathcal{C}$ before position $p(\beta)$; note that β occurs in $\mathcal{S}_{p(\beta),uv,|\beta|}$. We let $\text{Enc}_s(\beta) = (t, m)$ such that the t-th vertex in S_{visited} is the last vertex visited in β (that is v), and m is the position of β in $\mathcal{S}_{p(\beta),uv,|\beta|}$. Let $\beta_1, \ldots, \beta_T$ be the blocks of stale steps in the order they appear in $\mathcal{C}$. We store the concatenation of $\text{Enc}_s(\beta_1), \ldots, \text{Enc}_s(\beta_T)$, which we call S_{stale}.

We refer to the constructed $(P_{\text{fresh}}, P_{\text{boundary}}, P_{\text{stale}}, S_{\text{fresh}}, S_{\text{boundary}}, S_{\text{stale}})$ as the *encoding* of $\mathcal{C}$.

Unique reconstruction of linkage. In this part of the proof, we show that we can uniquely recover $\mathcal{C}$ from its encoding. First, since $\mathcal{C}$ is a canonical linkage we can correctly reconstruct S_{visited} from S_{fresh} because the labels are visited in canonical (increasing) order. From $P_{\text{fresh}}, P_{\text{boundary}}$ and P_{stale}, we can infer a partition of $[4k(\ell+1)]$ into blocks in order $\beta_1, \ldots, \beta_T$ and the type of each block. We sketch an inductive proof that shows how $\mathcal{C}$ can be uniquely recovered from its encoding. As our base case, the first block is a fresh block and hence all the steps that comprise it can be recovered from S_{visited}. Towards our inductive step, suppose we know the edges in $\mathcal{C}$ from blocks $\beta_1, \ldots, \beta_i$, we show how to recover the edges in β_{i+1} from the encoding of $\mathcal{C}$. If β_{i+1} is a fresh or boundary block, its recovery is straightforward. Suppose β_{i+1} is a stale block. Then from P_{stale} and S_{stale}, we can infer the last vertex v visited by β_{i+1} and the length of the block $|\beta_{i+1}|$. We know the first vertex u in β_{i+1} and can reconstruct $\mathcal{S}_{p(\beta_{i+1}),uv,|\beta_{i+1}|}$ since we have complete information about the steps in $\mathcal{C}$ prior to β_{i+1}. We can then infer β_{i+1} from $\mathcal{S}_{p(\beta_{i+1}),uv,|\beta_{i+1}|}$ and S_{stale}.

Bounding the number of metadata encodings. A fresh block must either be followed by a boundary step, or must occur at the end of a $2(\ell+1)$-segment; analogously, a stale block must either be preceded by a boundary step, or must occur at the start of a $2(\ell+1)$-segment. Thus, the number of fresh blocks and stale blocks are each bounded by $b + 2k$. Further, the number of boundary blocks is clearly bounded by b. Since there are at most $(4k(\ell+1))^2$ distinct combinations of starting and ending positions of a block, the number of distinct possibilities that the triple $(P_{\text{fresh}}, P_{\text{stale}}, P_{\text{boundary}})$ can be bounded by $(4k(\ell+1))^{6b+8k}$.

Bounding number of fresh step encodings. For a fixed P_{fresh}, we give an upper bound on the number of possibilities for S_{fresh}. Fixing P_{fresh} fixes a number T as well as $q_1, \ldots, q_T$ such that there are T fresh blocks in $\mathcal{C}$ and such that the i-th block has length q_i. Let us focus on a single fresh block β. The sequence of vertices in S_{fresh} corresponding to β give a nonbacktracking walk W_β in the base constraint graph $K_{r,c}$. Additionally, for a consecutive triple (i, j, i') in this nonbacktracking walk, $\{i, i'\}$ must be an edge in the corresponding base instance graph $\mathcal{I}_1$ due $\mathcal{C}$ being an $\mathcal{A}$-linkage. Let $\widetilde{W}_\beta$ be the maximal subwalk of W_β that starts and ends with a variable vertex. Note that $\widetilde{W}_\beta$ corresponds exactly to a nomadic walk in $\mathcal{I}_1$ whose length is at most $|\beta|/2$. Now regarding W_β, either W_β is equal to $\widetilde{W}_\beta$ (there is 1 way in which this can happen), or both the first and last steps of W_β are not in $\widetilde{W}_\beta$ (there are c^2 ways in which this can happen), or exactly one of the first and last steps of W_β is not in $\widetilde{W}_\beta$ (there are $2c$ ways in which this can happen). This tells us that the number of distinct possibilities for W_β is bounded by $(c+1)^2\delta_{\lfloor|\beta|/2\rfloor}$, where δ_s denotes the number of nomadic walks of length s in $\mathcal{I}_1$. Thus, we obtain an upper bound of $(c+1)^{2T}\prod_{i=1}^{T}\delta_{\lfloor q_i/2\rfloor}$ on the number of possibilities for S_{fresh}, which is bounded by $(c+1)^{2b+4k}\prod_{i=1}^{T}\delta_{\lfloor q_i/2\rfloor}$. Towards simplifying the expression, we bound δ_s. Observe that for a given edge $e \in E(|\mathcal{I}_1|)$, the number of nomadic walks of length s starting with e is given by $\|(|B_1|)^s\mathbf{1}_e\|_1$. This implies that $\delta_s \leqslant \|(|B_1|)^s\|_1$, where $\|(|B_1|)^s\|_1 = \sup\{\|(|B_1|)^s x\| : \|x\|_1 = 1\}$.

To bound the above, first observe that we have a simple bound $\|(|B_1|)^s\|_1 \leqslant \kappa^s$ provided κ is a large enough constant (for example, the maximum degree of $\mathcal{I}_1$ is a possible such value). Next, it is known that

$$\lim_{s\to\infty}\left(\|(|B_1|)^s\|\right)^{1/s} = \rho(|B_1|),$$

and hence for any $\hat{\rho} > \rho(|B_1|)$, there is a constant ℓ_0 such that $\|(|B|)^s\|_1 \leqslant (\hat{\rho})^s$ for all $s \geqslant \ell_0$. Putting these two bounds together we get that for any $s \geqslant \ell_0$,

$$\delta_s \leqslant \|(|B_1|)^s\|_1 \leqslant (\hat{\rho})^{s-\ell_0}\kappa^{\ell_0}.$$

Thus the number of possibilities for S_{fresh} is bounded by $(c+1)^{2b+4k}\prod_{i=1}^{T}(\hat{\rho})^{\lfloor q_i/2\rfloor-\ell_0}\kappa^{\ell_0}$, which can, in turn, be bounded by $\left((c+1)^2\kappa^{\ell_0}\hat{\rho}^{-\ell_0}\right)^{b+2k}(\hat{\rho})^{f/2}$.

Bounding number of stale step encodings. For any stale block β, let u and v be the first and last visited vertices respectively. S_{stale} specifies a number in $[f+1]$ to encode v, and a number between 1 and M where M is the total number of nonbacktracking walks from u to v of length $|\beta|$. Since the number of stale blocks is bounded by $b+2k$, the number of possibilities for what S_{stale} can be is at most $(M(f+1))^{b+2k}$. We show that $M \leqslant 2$, and hence translate our upper bound to $(2(f+1))^{b+2k}$.

Since all blocks are contained within $2(\ell+1)$-segments and the $\mathcal{A}$-linkage being encoded is 4ℓ-tangle-free, the steps traversed by β are in a connected subgraph H with at most one cycle. Our goal is to show that there are at most 2 nonbacktracking walks of a given length L between any pair of vertices x, y. There is at most one nonbacktracking walk between x and y that does not visit vertices on C, the single cycle in H, and if such a walk exists, it is the unique shortest path. Any nonbacktracking walk between x and y that visits vertices of C can be broken down

into 3 phases — (i) a nonbacktracking walk from x to v_x, the closest vertex in C to x, (ii) a nonbacktracking walk from v_x to v_y, the closest vertex in C to y, (iii) a nonbacktracking walk from v_y to y. Phases (i) and (iii) are always of fixed length, whose sum is some L'. Thus, it suffices to show that there are at most 2 nonbacktracking walks from v_x to v_y of length $L - L'$. Any nonbacktracking walk takes r rotations in C and then takes an acyclic path from v_x to v_y, whose length is observed to be strictly less than $|C|$, for $r \geqslant 0$. The steps in a nonbacktracking walk from v_x to v_y are either all in a clockwise direction, or all in an anticlockwise direction, and hence for any r there are at most 2 nonbacktracking walks from v_x to v_y of length strictly between $(r-1)|C|$ and $r|C|+1$. In particular, there are at most 2 nonbacktracking walks between v_x and v_y of length equal to $L - L'$.

Bounding number of boundary step encodings. S_{boundary} is a sequence of b tuples in $[f+1]^2$, and hence there are at most $(f+1)^{2b}$ distinct sequences that S_{boundary} can be.

Final bound: The above gives us a final bound of:

$$(4k(\ell+1))^{6b+8k}((c+1)^2\kappa^{\ell_0}(\hat{\rho})^{-\ell_0})^{b+2k}(\hat{\rho})^{f/2}2^{b+2k}(f+1)^{3b+2k} \tag{5.20}$$

which, when combined with Proposition 5.6.27 gives the desired claim. □

We wrap everything up by combining the results of Lemma 5.6.24 with Lemma 5.6.17 to prove Theorem 5.6.20.

Proof of Theorem 5.6.20. Let $\ell = \kappa \log n$, where κ is the universal constant from Proposition 5.6.5, let k be chosen so that $k\ell = \omega(\log n)$, let R be as in Lemma 5.6.17, and let $\hat{\rho}$ be any constant greater than $\rho(|B_1|)$. Then we have

$$\begin{aligned}
R &= \sum_{\substack{(2k\times 2\ell)\text{-nonbacktracking} \\ \text{valid, internally } \ell\text{-tangle-free, singleton-free} \\ \mathcal{A}\text{-linkages } \mathcal{C} \text{ in } \mathcal{K}_n}} n^{-|E(\mathcal{C})|} \\
&= \sum_{f=0}^{\infty}\sum_{b=0}^{\infty} |\mathrm{Lkgs}(f,b)| n^{-(f+b)} \\
&= \sum_{f=0}^{2k\ell+O(k\log(k\ell))}\sum_{b=0}^{\infty} |\mathrm{Lkgs}(f,b)| n^{-(f+b)} && \text{(by Corollary 5.6.19)} \\
&\leqslant \sum_{f=0}^{2k\ell+O(k\log(k\ell))}\sum_{b=0}^{\infty} \frac{\mathrm{poly}(k,\ell)^b \cdot \mathrm{poly}(k,\ell)^k \cdot (\hat{\rho})^{f/2}\cdot n}{n^b} && \text{(by Lemma 5.6.24)} \\
&= \sum_{f=0}^{2k\ell+O(k\log(k\ell))} n\cdot \mathrm{poly}(k,\ell)^k \cdot (\hat{\rho})^{f/2} \sum_{b=0}^{\infty}\left(\frac{\mathrm{poly}(k,\ell)}{n}\right)^b \\
&= \sum_{f=0}^{2k\ell+O(k\log(k\ell))} n\cdot \mathrm{poly}(k,\ell)^k \cdot (\hat{\rho})^{f/2} \cdot \left(\frac{1}{1-\frac{\mathrm{poly}(k,\ell)}{n}}\right)
\end{aligned}$$

$$\leqslant 2n \cdot \mathrm{poly}(k,\ell)^k (2k\ell + O(k\log(k\ell)))(\hat{\rho})^{k\ell + O(k\log(k\ell))}$$

For the choice of k and ℓ in the theorem statement, we can use Lemma 5.6.17 to conclude that

$$\rho(\boldsymbol{B}_n) \leqslant (1 + o_n(1)) \cdot \sqrt{\hat{\rho}}.$$

with probability $1 - O(n^{.99})$. Since the above bound holds for any $\hat{\rho} > \rho(|B_1|)$, for any $\varepsilon > 0$, it can be rewritten as

$$\rho(\boldsymbol{B}_n) \leqslant (1 + o_n(1)) \cdot (1+\varepsilon) \cdot \sqrt{\rho(|B_1|)}.$$ □

5.7 The SDP value for random two-eigenvalue CSPs

In this section, we put all the ingredients together to conclude our main theorem. We start with an elementary and well known fact and include a short proof for self containment.

Fact 5.7.1. *Let A be a real $n \times n$ symmetric matrix. Then*

$$\frac{1}{n} \max_{X \succeq 0, X_{ii}=1} \langle A, X \rangle \leqslant \lambda_{\max}(A)$$
$$\frac{1}{n} \min_{X \succeq 0, X_{ii}=1} \langle A, X \rangle \geqslant \lambda_{\min}(A)$$

Proof. We prove the upper bound below. The proof of the lower bound is identical.

$$\begin{aligned}\frac{1}{n} \max_{X \succeq 0, X_{ii}=1} \langle A, X \rangle &\leqslant \frac{1}{n} \max_{X \succeq 0, \mathrm{tr}(X)=n} \langle A, X \rangle \\ &= \max_{X \succeq 0, \mathrm{tr}(X)=1} \langle A, X \rangle \\ &= \lambda_{\max}(A).\end{aligned}$$

□

Recall $\alpha_{\mathrm{gr}} := (c-1)(-\lambda_1\lambda_2)$ and $r_X := 2\sqrt{\alpha_{\mathrm{gr}}}$.

Theorem 5.7.2. *Let $\mathcal{A} = (A_1, \ldots, A_c)$ be a sequence of r-vertex atoms with edge weights ± 1. Let $\boldsymbol{\mathcal{H}}_n$ denote a random n-lifted constraint graph and $\boldsymbol{\mathcal{I}}_n = \mathcal{A}(\boldsymbol{\mathcal{H}}_n)$ an associated instance graph with 1-wise uniform negations $(\boldsymbol{\xi}^f_{ii'})$. Let $\boldsymbol{A}_n$ be the adjacency matrix of $\boldsymbol{\mathcal{I}}_n$. Then, with probability $1 - o_n(1)$,*

$$\max_{X \succeq 0, X_{ii}=1} \langle \boldsymbol{A}_n, X \rangle = (\lambda_1 + \lambda_2 + r_X \pm \varepsilon)n$$
$$\min_{X \succeq 0, X_{ii}=1} \langle \boldsymbol{A}_n, X \rangle = (\lambda_1 + \lambda_2 - r_x \pm \varepsilon)n.$$

Proof. $\max_{X \succeq 0, X_{ii}=1} \langle \boldsymbol{A}_n, X \rangle \geqslant (\lambda_1+\lambda_2+r_X-\varepsilon)n$ follows from Theorem 5.5.2 and $\max_{X \succeq 0, X_{ii}=1} \langle \boldsymbol{A}_n, X \rangle \leqslant (\lambda_1+\lambda_2+r_X+\varepsilon)n$ follows from Fact 5.7.1. The upper and lower bounds on $\min_{X \succeq 0, X_{ii}=1} \langle \boldsymbol{A}_n, X \rangle$ can be determined identically. □

Chapter 6

Girth and Ramanujan Graphs

In this Chapter we describe the results of [Par21]. In this paper we described a new method to remove short cycles on regular graphs while maintaining spectral bounds (the nontrivial eigenvalues of the adjacency matrix), as long as the graphs have certain combinatorial properties. These combinatorial properties are related to the number and distance between short cycles and are known to happen with high probability in uniformly random regular graphs.

Using this method we were able to show two results involving high girth spectral expander graphs: there exists an explicit distribution of d-regular $\Theta(n)$-vertex graphs where with high probability its samples have girth $\Omega(\log_{d-1} n)$ and are ε-near-Ramanujan; there is a a deterministic $\mathrm{poly}(n)$-time algorithm that outputs a d-regular graph on $\Theta(n)$-vertices that is ε-near-Ramanujan and has girth $\Omega(\sqrt{\log n})$.

6.1 Regular graphs and short cycles

The study of short cycles and *girth* (defined as the length of the shortest cycle of a graph) in such graphs dates back to at least the 1963 paper of Erdős and Sachs [ES63], who showed that there exists an infinite family with girth at least $(1 - o_n(1)) \log_{d-1} n$. On the converse side, a simple path counting argument known as the "Moore bound" shows that this girth is upper bounded by $(1 + o_n(1))2 \log_{d-1} n$. Though simple, this is the best known upper bound. Given these bounds, it is common to call an infinite family of d-regular n-vertex graphs *high girth* if their girth is $\Omega(\log_{d-1} n)$.

The first explicit construction of high girth regular graphs is attributed to Margulis [Mar82], who gave a construction of graphs that achieve girth $(1-o_n(1))\frac{4}{9} \log_{d-1} n$. A series of works initiated by Lubotzky-Phillips-Sarnak [LPS88] and then improved by several other people [Mar88, Mor94, LU95] culminated in the work of Dahan [Dah14], who proves that for all large enough d there are explicit d-regular n-vertex graphs of girth $(1 - o_n(1))\frac{4}{3} \log_{d-1} n$.

Another relevant problem consists of generating random distributions that produce regular graphs with high girth. Results regarding the probabilistic aspects of certain structures (like cycles) in graphs often give us tools to count the number of graphs that satisfy certain conditions, like how many regular graphs have girth at least some value. The distribution of short cycles in uniformly random regular graphs was first studied by Bollobás [Bol80], who proved, that

for a fixed k the random variables representing the number of cycles of length exactly k in a uniformly random d-regular graph are asymptotically independent Poisson with mean $(d-1)^k/2k$. Subsequently, McKay-Wormald-Wysocka [MWW04] gave a more precise description of this by finding the asymptotic probability of a random d-regular graph having a certain number of cycles of any length up to $c\log_{d-1} n$, for $c < 1/2$. More recently, Linial and Simkin [LS21] showed that a random greedy algorithm that is given $d \geqslant 3$, $c \in (0,1)$ and an even n, produces a d-regular n-vertex graph with girth at least $c\log_{d-1} n$ with high probability.

The literature of regular graphs with high girth is closely connected to the literature of spectral expanders, which was the focus of Chapter 4. In this Chapter we concern ourselves with bridging these two worlds, looking for families of regular graphs that are both good spectral expanders and also have high girth. This bridge can be seen in several works. The explicit construction of high girth regular graphs by Margulis [Mar82] was a motivator to his work on Ramanujan graphs [Mar88]. Additionally, the constructions of [LPS88] and [Mor94] produce graphs that are both Ramanujan and have girth $(1-o_n(1))\frac{4}{3}\log_{d-1} n$, according to the previously stated restrictions on d.

More recently, Alon-Ganguly-Srivastava [AGS21] showed that for a given d such that $d-1$ is prime and $\alpha \in (0, 1/6)$, there is a construction of infinite families of graphs with girth at least $(1-o_n(1))(2/3)\alpha\log_{d-1} n$ and λ at most $(3/\sqrt{2})\sqrt{d-1}$ with many eigenvalues localized on small sets of size $O(n^\alpha)$. Their motivation comes from the theory of quantum ergodicity in graphs, which relates high-girth expanding graphs to delocalized eigenvectors. See [AGS21] for more on this. Our main result is based on some of the techniques of this work.

One other motivation to search for graphs with simultaneous good spectral expansion and high girth is its application to the theory of error-correcting codes, particularly for *Low Density Parity Check* or *LDPC* codes, originally introduced by Gallager [Gal62]. The connection with high girth regular graphs was first pointed out by Margulis in [Mar82]. The property of high-girth is desirable since the decoding of such codes relies on an iterative algorithm whose performance is worse in the presence of short cycles. Additionally, using graphs with good spectral properties to generate these codes heuristically seems to lead to good performance, as pointed out by several works [RV00, LR00, MS02].

6.1.1 Our results

We can now state our results and put them in perspective. Let's first introduce some useful definitions and notation.

Definition 6.1.1 ((r,τ)-graph)**.** Let r and τ be a positive integers. Then, we call a graph G a (r,τ)-graph if it satisfies the following conditions:

- G is bicycle-free at radius at least r;
- The number of cycles of length at most r is at most τ.

Our main result is the following short cycle removal theorem:

Theorem 6.1.2. *There exists a deterministic polynomial-time algorithm* fix *that, given as input a d-regular n-vertex (r,τ)-graph G satisfying $r \leqslant (2/3)\log_{d-1}(n/\tau) - 5$ outputs a graph* $\mathrm{fix}(G)$ *satisfying*

- $\mathrm{fix}(G)$ *is a d-regular graph with $n + O(\tau\cdot(d-1)^{r/2+1})$ vertices;*

- $\lambda(\mathrm{fix}(G)) \leqslant \max\{\lambda(G), 2\sqrt{d-1}\} + O_d(1/r)$*;*
- $\mathrm{fix}(G)$ *has girth at least* r*.*

The key fact in our proof of this statement is a theorem proved by Kahale [Kah95], originally used to construct Ramanujan graphs with better expansion of sublinear sized subsets. See also [AGS21] and [Alo21] for other applications of this technique. We will prove this theorem in Section 6.2.

The preconditions of this theorem are not arbitrary. Even though random uniformly n-vertex d-regular graphs have constant girth with high probability, they are bicycle-free at radius $\Omega(\log_{d-1} n)$ and the number of cycles of length at most $c \log_{d-1} n$ (for small enough c) is $o(n)$ with high probability. Recall that from Theorem 1.1.16 we also know that being near-Ramanujan is also a property that occurs with high probability in random regular graphs. So a statement like the above can be used to produce distributions over regular graphs that have high girth and are near-Ramanujan with high probability. With this in mind, we introduce the following definition:

Definition 6.1.3. ((Λ, g)-good graphs). We call a graph G a (Λ, g)-good graph if $\lambda(G) \leqslant \Lambda$ and $\mathrm{girth}(G) \geqslant g$.

Let $\mu_d(n)$ be a distribution over d-regular graphs with $\sim n$ vertices. We say $\mu_d(n)$ is (Λ, g)-good if $G \sim \mu_d(n)$ is (Λ, g)-good with probability at least $1 - o_n(1)$.

Additionally, we call the distribution explicit if sampling an element is doable in polynomial time.

We shall prove the following using Theorem 6.1.2 in Section 6.3:

Theorem 6.1.4. *Given $d \geqslant 3$ and n, let G be a uniformly random d-regular n-vertex graph. For any $c < 1/4$ and $\epsilon > 0$, $\mathrm{fix}(G)$ is a $(2\sqrt{d-1} + \epsilon, c \log_{d-1} n)$-good explicit distribution.*

Recall that the upper bound on the girth of a regular graph is $(1 + o_n(1))2 \log_{d-1} n$, so this distribution has optimal girth up to a constant. Based on our proof of the above and using some classic results about the number of d-regular n-vertex graphs, we can show a lower bound on the number of $(2\sqrt{d-1} + \epsilon, c \log_{d-1} n)$-good graphs in some range.

Corollary 6.1.5. *Let $d \geqslant 3, n$ be integers and $\epsilon > 0, c > 1/4$ reals. The number of d-regular graphs with number of vertices in $[n, n + O(n^{3/8})]$, which are $(2\sqrt{d-1} + \epsilon, c \log_{d-1} n)$-good, is at least*

$$\Omega\left(\left(\frac{d^d n^d}{e^d (d!)^2}\right)^{n/2}\right).$$

We prove both of these results in Section 6.3.

Finally, we show a slightly stronger version of result of [MOP20a] by plugging our short cycle removal theorem into their construction.

Theorem 6.1.6. *Given any integer n and constants $d \geqslant 3$, $\epsilon > 0$ and c, there is a deterministic polynomial-time (in n) algorithm that constructs a d-regular N-vertex graph with the following properties:*

- $N = n(1 + o_n(1))$*;*
- $\lambda(G) \leqslant 2\sqrt{d-1} + \epsilon$*;*
- *G has girth at least $c\sqrt{\log n}$.*

Note that this only works for large enough n. Also, the running time from the theorem above has an exponential dependency on d, ϵ and c. The proof of this statement as well as the precise dependencies on these constants will be worked out in Section 6.4.

6.2 Short cycles removal

In this section we prove Theorem 6.1.2. Recall that we are given a d-regular n-vertex (r, τ)-graph G with the constraint specified in Theorem 6.1.2 and we wish to find some d-regular graph $\mathrm{fix}(G)$ on $\sim n$ vertices such that $\lambda(\mathrm{fix}(G)) \leqslant \lambda(G) + o_r(1)$ and its girth is at least r.

Briefly, the algorithm that achieves this works by removing one edge per small cycle from G, effectively breaking apart all such cycles, and then fixing the resulting off degree vertices by adding d-ary trees in a certain way. We will now more carefully outline this method and then proceed to fill in some details as well as show it works as desired.

Before starting, we introduce some notation which will be helpful.

Definition 6.2.1 ($\mathrm{Cyc}_g(G)$)**.** Given a graph G, let $\mathrm{Cyc}_g(G)$ denote the collection of all cycles in G of length at most g. Recall that if $\mathrm{Cyc}_g(G)$ is empty then G is said to have girth exceeding g.

Definition 6.2.2 ($B_\delta(S)$)**.** Given a set of vertices S in a graph G, let $B_\delta(S)$ denote the collection of vertices in G within distance δ of S. We will occasionally abuse this notation and write $B_\delta(v)$ instead of $B_\delta(\{v\})$ for a vertex v.

Let E_c be a set containing exactly one arbitrary edge per cycle in $\mathrm{Cyc}_r(G)$. Note that the bicycle-freeness property implies E_c is a matching. Let H_t be a graph with the same vertex set as G obtained by removing all edges in E_c from G. To prevent ambiguity, whenever we pick something arbitrarily let's suppose the algorithm fix uses the lexicographical order of node labels as a tiebreaker. We also partition the endpoints of each edge as described in the following definition:

Definition 6.2.3 ($V_i(E)$)**.** Given a matching E, we let $V_1(E)$ and $V_2(E)$ be two disjoint sets of vertices constructed as follows: for all $e = (u, v) \in E$ place u in $V_1(E)$ and v in $V_2(E)$ (so each endpoint is in exactly one of the two sets).

Note that according to the above definition we have $|V_1(E_c)| = |V_2(E_c)| = |E_c| \leqslant \tau$. For ease of notation we also define:

Definition 6.2.4 ($\phi_E(v)$)**.** Given a matching E and $(u, v) \in E$ such that $u \in V_1(E)$ and $v \in V_2(E)$, we denote by ϕ_E the function that maps endpoints to endpoints, so we have $\phi_E(u) = v$ and $\phi_E(v) = u$.

We will often abuse notation and drop the E from ϕ_E when it is clear from context.

Since we break apart each cycle in $\mathrm{Cyc}_r(G)$, we can conclude that H_t has girth greater than r. However, note that in removing edges, H_t is no longer d-regular.

To fix this, consider the following object which we refer to as a d-regular tree of height h: a finite rooted tree of height h where the root has d children but all other non-leaf vertices have $d-1$ children. This definition implies that every non-leaf vertex in a d-regular tree has degree d.

We shall add two d-regular trees to H_t in order to fix the off degrees, while maintaining the desired girth and bound on λ. The idea of using d-regular trees is based on the degree-correction gadget used in [AGS21] for their construction of high-girth near-Ramanujan graphs

with localized eigenvectors. As such, we will use some of the tools used in their proofs.

Let h be an integer parameter we shall fix later. Let T_1 and T_2 be two d-regular trees of height h and let L_1 and L_2 be the sets of leaves of each one. Note that $|L_1| = |L_2| = d(d-1)^{h-1} \approx (d-1)^h$. We shall add the two trees to H_t and then pair up elements of $V_1(E_c)$ with elements of L_1 (and analogously for $V_2(E_c)$ and L_2) and merge the paired up vertices. However, we have to deal with two potential issues:

- $|L_i| \neq |V_i(E_c)|$, in which case we cannot get an exact pairing between these sets;
- This procedure might result in the creation of small cycles (potentially even cycles of length $O(1)$).

To expand on the latter point, we describe a potential problematic instance. Suppose we can somehow pick h such that $|L_i| = |V_i(E_c)|$ and then arbitrarily pair up their elements. Suppose there are two edges in E_C corresponding to two cycles of constant length and denote their endpoints by $v_1 \in V_1(E_C), v_2 \in V_2(E_C)$ and $u_1 \in V_1(E_C), u_2 \in V_2(E_C)$. If the distance in T_1 of v_1 and u_1 given by the pairing of $V_1(E_c)$ and L_1 is small (constant, for example) and the same applies to the distance in T_2 of v_2 and u_2, then there is a cycle of small length (constant, for example) in the graph resulting from adding the two trees to H_t.

To address this issue we remove some extra edges from G that are somehow "isolated" and group them with edges from E_C. The goal is to have the endpoints of any two edges in E_C be far apart in T_1 and T_2 distance, but close to some of the endpoints of the extra edges. With this in mind, we set $h = \lceil \log_{d-1} \tau \rceil + \lceil r/2 \rceil + 1$ so that $|L_i| \approx \tau \cdot (d-1)^{r/2+1}$, which is close to the number of extra edges we want to remove. This choice will also be helpful later when we analyze the spectral properties of the construction.

Formally, this leads us to the following proposition:

Proposition 6.2.5. *There is a set of edges E_t of G such that the following is true for $i \in \{1, 2\}$:*

- $|V_i(E_t) \cup V_i(E_c)| = d(d-1)^{h-1}$*;*
- *for all distinct $u, v \in V_i(E_t) \cup V_i(E_c)$, we have $v \notin B_r(u)$ and $u \notin B_r(v)$.*

Additionally, we can find such a set in polynomial time.

Proof. We will describe the efficient algorithm that does this.

We are going to incrementally grow our set E_t, one edge at the time, until $|V_i(E_t) \cup V_i(E_c)| = d(d-1)^{h-1}$, so suppose E_t is initially an empty set. We start by, for all $e = (u, v) \in E_c$, marking all vertices in $B_{1+r}(\{v, u\})$. Note that we marked at most $\tau \cdot (d(d-1)^r) \leqslant 2\tau(d-1)^{r+1}$ vertices.

Notice that, since we marked all vertices at distance $1 + r$ from any vertex in $V_i(E_c)$, we can safely pick any unmarked vertex and an arbitrary neighbor and add that edge to E_t.

We can now describe a procedure to add a single edge to E_t:

- Pick an unmarked vertex u and an arbitrary neighbor v of u;
- Add (u, v) to E_t;
- Mark all vertices in $B_{1+r}(\{u, v\})$.

By the same reasoning as before, as long as we have an unmarked vertex, this procedure works. If we repeat the above t times, we are left with at least $n - 2\tau(d-1)^{r+1} - 2t(d-1)^{r+1}$

unmarked vertices. We claim the procedure can be successfully repeated at least $2\tau(d-1)^{r/2+2}$ times. In such a case, the number of unmarked vertices left is at least:

$$n - 2\tau(d-1)^{r+1} - 4\tau(d-1)^{r/2+2}(d-1)^{r+1} \geqslant n - 6\tau(d-1)^{3r/2+3},$$

which is always greater than 0 when $r \leqslant \frac{2}{3}\log_{d-1}(n/\tau) - 5$. Hence, we always have at least one unmarked vertex to pick throughout the procedure.

Note that the number of repetitions we require exactly matches the size of $|E_t|$ so we need this to be exactly $d(d-1)^{h-1} - \tau \leqslant 2\tau(d-1)^{r/2+2}$, which means our algorithm always succeeds. □

We will state some simple properties of this construction that will be relevant later on.

Fact 6.2.6. $|V_i(E_t)| \geqslant \tau \cdot (d-1)^{\lceil r/2 \rceil}$

Proof. We simply have: $|V_i(E_t)| = |E_t| = d(d-1)^{h-1} - \tau \geqslant \tau \cdot (d-1)^{\lceil r/2 \rceil}$. □

Fact 6.2.7. *For all $e \in E_t$, there is at most one cycle in $B_r(e)$ in G and if there is a cycle it has length greater than r.*

Proof. That there is at most one cycle in $B_r(e)$ is obvious since G is bicycle-free at radius r. So, let's suppose there is a cycle C in $B_r(e)$ with length less than or equal to r. Then, there is at least one edge $e' \in C$ that is also in E_c, but in that case $e' \in B_r(e)$, which contradicts the definition of E_t. □

We can now extend our definition of H_t. Let H be the graph obtained from G by removing all edges in E_c and in E_t.

Recall our plan to add T_1 and T_2, two d-regular trees of height h (recall $h = \lceil \log_{d-1} \tau \rceil + \lceil r/2 \rceil + 1$), to H while pairing up elements of L_i with endpoints of removed edges. We will now describe a pairing process that achieves high girth (and later we will see how it also achieves low λ).

First, consider a canonical ordering of L_1 and L_2 based on visit times from a breath-first search, as illustrated in Figure 6.1 for $d = 3$. Given this ordering, the following is easy to see:

Fact 6.2.8. *The tree distance between two leaves with indices i and j is at least* $2(1+\log_{d-1}((|i-j|+1)/d))$.

Proof. Let's show that the lowest common ancestor of the two leaves is at least $1 + \log_{d-1}((|i-j|+1)/d)$, this proves the claim since we need to travel this distance twice, from the ith indexed leaf to the ancestor and then back to the jth indexed leaf. Let V_0 be the set of $|i-j|+1$ leaves with indices between i and j. Let's construct the smallest subtree that includes V_0 from bottom up and compute its height, which is an upper bound to the desired lowest common ancestor. First, group elements of V_0 in groups of at most $d-1$ consecutive indices and add one representative of each group to a set V_1. Each group corresponds to a node that parents all of its elements. There are at most $|V_0|/(d-1)$ such groups, so $|V_1| \leqslant |V_0|/(d-1)$. Repeat the same procedure until $|V_a| \leqslant 1$, in which case a is an upper bound to the height of the goal subtree, and by induction we have that $|V_{i+1}| \leqslant |V_i|/(d-1)$, so $a \geqslant \log_{d-1}|V_0|$.

This is not quite right because if the last grouping corresponds to the root of the tree, we need to group elements in d groups, because this is the degree of the root, so by accounting for this we have $a \geqslant 1 + \log_{d-1}(|V_0|/d)$. □

Now, consider the following pairing of elements in L_1 and $V_1(E_t) \cup V_1(E_c)$: pick an arbitrary element of $V_1(E_c)$ and pair it up with the first leaf of L_1. Now pick $(d-1)^{\lceil r/2 \rceil}$ distinct elements of $V_1(E_t)$ and pair them up with the next leaves of L_1. Repeat this procedure, of pairing one element of $V_1(E_c)$ with $(d-1)^{\lceil r/2 \rceil}$ elements of $V_1(E_t)$ with a contiguous block of leaves until we exhaust all elements of $V_1(E_c)$. Note that by Fact 6.2.6, there always are enough elements in E_t to perform this pairing. Pair up any remaining leaves with the remaining elements of $V_1(E_t)$ arbitrarily. Now repeat the same procedure but for L_2 and $V_2(E_t) \cup V_2(E_c)$ with the same groupings (so the endpoints of an edge in either E_t or E_c are mapped to the same leaves of L_1 and L_2). This pairing procedure is pictured in Figure 6.2 below.

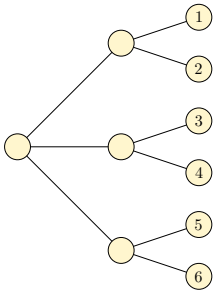

Figure 6.1: Leaf ordering for $d = 3$

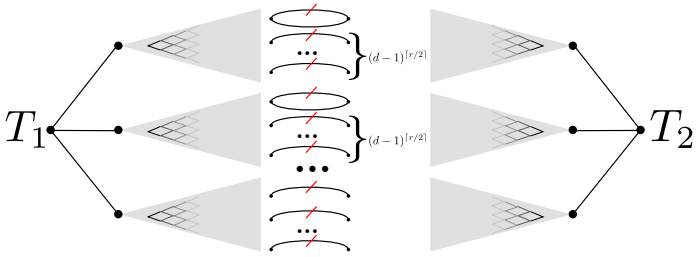

Figure 6.2: Example pairing

Let $\mathrm{fix}(G)$ be defined as the graph resulting from applying the method described in the previous paragraph to fix the degrees of H. It is now obvious that $\mathrm{fix}(G)$ is a d-regular graph and we only add $|T_1| + |T_2| = O(\tau \cdot (d-1)^{r/2+1})$ new vertices, so it has $n + O(\tau \cdot (d-1)^{r/2+1})$ total vertices. We will now analyze the resulting girth and λ value and prove Theorem 6.1.2 in the process.

6.2.1 Analyzing the girth of $\mathrm{fix}(G)$

Here we prove that the girth of $\mathrm{fix}(G)$ is at least r. Let's start by supposing, for the sake of contradiction, that there is a cycle C of length less than r. We know that the girth of H is more than r by definition, so C has to use an edge from T_1 or T_2. Without loss of generality, let's assume that C contains at least one edge from T_1. Since T_1 is a tree, C has to eventually exit T_1 and use some edges from H, so in particular it uses some vertex $v \in L_1$. We will show that in this case C has length at least r, which is a contradiction. Thus, we have to handle two cases: $v \in V_1(E_c)$ and $v \in V_1(E_t)$.

Let us start with the $v \in V_1(E_c)$ case. Let's follow C starting in v and show that to loop back to v, C would require to traverse at least r edges. So, we start in v and go into T_1 by following the only edge in T_1 that connects to v. Then, the cycle C has to use some edges from T_1 and finally exit through some other vertex in L_1 before eventually looping back to v. Suppose that $u \in L_1$ is such a vertex. Due to our grouping of elements in E_t with $(d-1)^{\lceil r/2 \rceil}$ elements in E_c, if u is in $V_1(E_c)$, we know that the tree indices of v and u differ by at least $(d-1)^{\lceil r/2 \rceil}$. Hence, plugging this into the bound from Fact 6.2.8, the tree distance between v and u is at least $r-1$, which would imply C has length at least r. So u has to be in $V_1(E_t)$.

Continuing our traversal of C, we now exit T_1 through u and need to loop back to v. From our construction in Proposition 6.2.5 we know that the distance in H between v and u is at least r, so any short path in $\mathrm{fix}(G)$ between these vertices has to go through T_1 or T_2. Again, our Proposition 6.2.5 construction gives that the distance in H between v and any other vertex in L_1 is at least r, so such a short path will have to use some edges in T_2.

Finally, we claim that the distance from u to any vertex w in L_2 is at least r. If $w \neq \phi(u)$, we know from our Proposition 6.2.5 construction that the distance between u and w is at least r. Otherwise, if there is a path P of length less than r from u to w, then the cycle $P + uw$ has length at most r and is in $B_r(\{u, w\})$, which contradicts Fact 6.2.7. In conclusion, it is not possible to loop back to v using less than r steps, which concludes the proof of the $v \in V_1(E_c)$ case.

The proof for the $v \in V_1(E_t)$ case is already embedded in the previous proof, so we will just sketch it. Using the same argument we start by following C into T_1 and eventually exiting through some vertex $u \in V_1(E_t)$. As we saw before, the H distance between u and v is at least r and the H distance between u and any other vertex in L_1 or any vertex in L_2 is at least r, so we cannot loop back to v from u, which concludes the proof of this case.

6.2.2 Bounding $\lambda(\mathrm{fix}(G))$

We finally analyze the spectrum of $\mathrm{fix}(G)$ by proving that $\lambda(\mathrm{fix}(G)) \leqslant \lambda(G) + O_d(1/r)$. This argument is similar to the proof in Section 4 of [AGS21], but adapted to our construction.

First, observe that the adjacency matrix of $\mathrm{fix}(G)$, which we will denote by simply A, can be written in the following way: $A = A_G - A_{E_c} - A_{E_t} + A_{T_1} + A_{T_2}$, where A_G is the adjacency matrix of G defined on the vertex set of $\mathrm{fix}(G)$ (which is to say G with a few isolated vertices from the added trees), A_{E_c} is the adjacency matrix of the cycle edges removed, and so on. Also, let V_G be the set of vertices from G, V_1 the set of vertices from T_1 and V_2 the set of vertices from T_2, so $V = V_G \cup V_1 \cup V_2$. In this section we will prove $\lambda(A) \leqslant \lambda(G) + O_d(1/r)$.

Let g be any unit eigenvector of A orthogonal to the all ones vector, so $\sum_{v \in V} g_v^2 = 1$ and $\sum_{v \in V} g_v = 0$. We have that $|\sum_{v \in V_1 \cup V_2} g_v| \leqslant \sqrt{2|T_i|}$ by Cauchy-Schwarz (since this vector is supported on only $2|T_i|$ entries), which in turn implies that $|\sum_{v \in V_G} g_v| \leqslant \sqrt{2|T_i|}$.

It suffices to show that $|g^T A g| \leqslant \lambda(G) + O_d(1/r)$. To do so, we shall analyze the contributions of A_G, A_{E_c}, A_{E_t}, A_{T_1} and A_{T_2} to $|g^T A g|$.

To bound the contribution of A_{T_1} and A_{T_2}, we use a lemma proved by Alon-Ganguly-Srivastava:

Lemma 6.2.9. *([AGS21, Lemma. 4.1]). Let W_i be the set of non-leaf vertices of T_i. Then for any vector f we have:*

$$|f^T A_{T_i} f| \leqslant 2\sqrt{d-1} \sum_{w \in W_i} f_w^2 + \sqrt{d-1} \sum_{v \in L_i} f_v^2.$$

Recall that the edges in $E_t \cup E_c$ define a perfect matching between L_1 and L_2, so we have the following:

$$|g^T (A_{E_c} + A_{E_t}) g| = \left| \sum_{uv \in E_t \cup E_c} 2 g_u g_v \right| \leqslant \sum_{v \in L_1 \cup L_2} g_v^2.$$

Finally, let g_G be the projection of g to the subspace spanned by V_G. Observe that $|g^T A_G g| = |g_G^T A_G g_G|$. Now, let $\mathbf{1}_G$ be the all ones vector supported on the set V_G and $g_\perp$ be a vector orthogonal to $\mathbf{1}_G$ such that $g_G = a\mathbf{1}_G + g_\perp$, for some constant a. We have that $\mathbf{1}_G^T g_G = a\mathbf{1}_G^T \mathbf{1}_G$, which implies

$$|a| = \left| \frac{\sum_{v \in V_G} (g_G)_v}{n} \right| \leqslant \frac{\sqrt{2|T_i|}}{n}.$$

Now observe:

$$|g_G^T A_G g_G| \leqslant |g_\perp^T A_G g_\perp| + |(a\mathbf{1}_G)^T A_G (a\mathbf{1}_G)| \leqslant \lambda(G) \sum_{v \in V_G} g_v^2 + \frac{2|T_i|d}{n}.$$

Note that $\sum_{v \in V_G} g_v^2 \leqslant 1$. We claim that the term $\frac{2|T_i|d}{n}$ is $O_d(1/r)$. We have $|T_i| = O(\tau \cdot (d-1)^{r/2+1})$ and we know from the problem constraints that $r \leqslant (2/3)\log_{d-1}(n/\tau) - 5$ which implies $\tau \cdot (d-1)^{r/2+1}/n \leqslant O((d-1)^{-r}) = O_d(1/r)$.

We can now plug everything together and apply Lemma 6.2.9 to obtain:

$$|g^T A g| \leqslant \lambda(G) + (\sqrt{d-1} + 1) \sum_{v \in L_1 \cup L_2} g_v^2 + O_d(1/r).$$

We will conclude our proof by showing that $\sum_{v \in L_1 \cup L_2} g_v^2$ is $O(1/r)$. It should be clear from the symmetry of our construction that we only need to prove $\sum_{v \in L_1} g_v^2 = O(1/r)$, since the same is analogous for L_2.

The following lemma can be proved using a known method by Kahale [Kah95, Lemma 5.1]. This statement is similar to one found in [Alo21, Lemma 3.2] and its proof is also very similar. For completeness, we present a self-contained proof of that based on the one from [Alo21].

Lemma 6.2.10. *Let v be some vertex of V. Let l be a positive integer such that $B_l(v)$ forms a tree. Let X_i be the set of all vertices at distance exactly i from v in $\mathrm{fix}(G)$, so $X_0 = \{v\}$. Let f be any non zero eigenvector with eigenvalue $|\mu| \geqslant 2\sqrt{d-1}$. Then, for $1 \leqslant i \leqslant l$:*

$$\sum_{u \in X_i} f^2(u) \geqslant \sum_{u \in X_{i-1}} f^2(u)$$

Proof. We will proceed by induction on i. First of all, let's establish the $i = 1$ case. Note that we have $\sum_{u \in X_1} f(u) = \mu f(v)$.

By Cauchy-Schwarz we get $d \cdot \sum_{u \in X_1} f^2(u) \geqslant \mu^2 f^2(v)$, and using the fact that $|\mu| \geqslant 2\sqrt{d-1}$ we obtain the desired:

$$\sum_{u \in X_1} f^2(u) \geqslant \frac{\mu^2}{d} f^2(v) \geqslant f^2(v).$$

Let's now assume that the statement is true for $i-1$ and prove that this implies it is true for i. Let u be some vertex in X_{i-1}. Recall that $B_l(v)$ is a tree and let u' be its parent in X_{i-2} and $w_1, \ldots w_{d-1}$ be its children in X_i. We have $f(u') + \sum_{i=1}^{d-1} f(w_i) = \mu f(u)$. Note that $f(u') = \sqrt{d-1} f(u')/\sqrt{d-1}$ and apply Cauchy-Schwarz to obtain:

$$\left(\frac{f^2(u')}{d-1} + \sum_{i=1}^{d-1} f^2(w_i)\right)(2d-2) \geqslant \mu^2 f^2(u),$$

which implies

$$\frac{f^2(u')}{d-1} + \sum_{i=1}^{d-1} f^2(w_i) \geqslant \frac{\mu^2}{2d-2} f^2(u) \geqslant 2f^2(u),$$

where the last inequality follows from the fact that $|\mu| \geqslant 2\sqrt{d-1}$.

We can finally sum the above for all $u \in X_{i-1}$, noting that from the fact that $B_l(v)$ is a tree we know that each element in X_{i-2} appears $d-1$ times (as the parent of $d-1$ vertices) and each element in X_i appears once:

$$\sum_{u \in X_{i-2}} f^2(u) + \sum_{u \in X_i} f^2(u) \geqslant 2 \sum_{u \in X_{i-1}} f^2(u).$$

We now apply the induction hypothesis and obtain the result:

$$\sum_{u \in X_i} f^2(u) \geqslant 2 \sum_{u \in X_{i-1}} f^2(u) - \sum_{u \in X_{i-2}} f^2(u) \geqslant \sum_{u \in X_{i-1}} f^2(u).$$

□

Our plan is to pick the parameters l and v from Lemma 6.2.10 and use it to show that $\sum_{v \in L_1} g_v^2 = O(1/r)$. Let μ be the eigenvalue associated with g and suppose that $|\mu| > 2\sqrt{d-1}$, otherwise $|\mu| \leqslant \lambda(G)$, which would imply the result. Set v to be the root of T_1. We will show that if we pick $l = h + \lfloor r/2 \rfloor$, where $h = \lceil \log_{d-1} \tau \rceil + \lceil r/2 \rceil + 1$ is the height of T_1 and T_2, then $B_l(v)$ forms a tree.

Note that $B_h(v)$ is exactly T_1, so it obviously forms a tree. To observe what happens in $B_l(v) \setminus B_h(v)$, we first prove the following proposition, whose proof uses some of the ideas of Section 6.2.1:

Proposition 6.2.11. *Let u be a vertex in L_1. Let $\mathcal{P}(u)$ be the set of non-empty paths that start in u and whose first step does not go into T_1. Then, the shortest path in $\mathcal{P}(u)$ that ends in any vertex in L_1 has length at least r.*

Proof. As in the previous girth proof, we have two cases, $u \in V_1(E_c)$ and $u \in V_1(E_t)$. The latter case is obvious from the proof in Section 6.2.1, since if $u \in V_1(E_t)$ then the H distance to any node in L_1 is at least r (from Proposition 6.2.5) and the H distance to any node in L_2 is also at least r (from Fact 6.2.7). So, suppose $u \in V_1(E_c)$.

Let's follow the same proof strategy as before, so let $P \in \mathcal{P}(u)$ be the shortest path and let's follow P starting in u. Again, from Proposition 6.2.5 the H distance of u to any node in L_1 is at least r. However, u might reach $\phi(u)$ in a short number of steps (namely, if the cycle corresponding to $(u, \phi(u))$ is short). So, let's follow P to $\phi(u)$ and into T_2. We are now in the exact same situation as in the setup of the proof in Section 6.2.1 (but starting in T_2), so the result follows. □

Let u be some vertex in L_1. Let's say a vertex w is at $\mathcal{P}$-distance δ from u if the shortest path $P \in \mathcal{P}(u)$ that ends in w has length δ. Additionally, let $S_\delta(u)$ be the set of vertices that are at a $\mathcal{P}$-distance of at most δ from u. From Proposition 6.2.11, we know that for all distinct $u, w \in L_1$, the sets $S_{\lfloor r/2 \rfloor}(u)$ and $S_{\lfloor r/2 \rfloor}(w)$ are disjoint. Thus, we have that for $u \in L_1$ the vertices in $S_{\lfloor r/2 \rfloor}(u)$ form disjoint trees rooted at u, which shows that $B_l(v)$ forms a tree.

We can now apply Lemma 6.2.10 and conclude that for all $1 \leqslant i \leqslant l$, we have $\sum_{u \in X_i} g_u^2 \geqslant \sum_{u \in X_{i-1}} g_u^2$. So the sequence $(\sum_{u \in X_i} g_u^2)_i$ is an increasing sequence. Note that $X_h = L_1$, so $\sum_{u \in X_h} g_u^2 = \sum_{u \in L_1} g_u^2$. Additionally, we know that the total sum of $(\sum_{u \in X_i} g_u^2)_i$ is at most one (since g is a unit vector and the X_i are disjoint), so we have that $\lfloor r/2 \rfloor \cdot \sum_{u \in X_h} g_u^2 \leqslant \sum_{i=h}^{l} \sum_{u \in X_i} g_u^2 \leqslant 1$ and finally $\sum_{u \in L_1} g_u^2 = \sum_{u \in X_h} g_u^2 \leqslant 1/\lfloor r/2 \rfloor = O(1/r)$.

This concludes the proof of Theorem 6.1.2.

6.3 A near-Ramanujan graph distribution of girth $\Omega(\log_{d-1} N)$

Recall Theorem 1.1.16, which says that uniformly random d-regular graphs are near-Ramanujan. We will combine this result with our machinery of Section 6.2 to show Theorem 6.1.4, namely that there exists a distribution over graphs that is $(2\sqrt{d-1} + \epsilon, c \log_{d-1} n)$-good for any $\epsilon > 0$ and $c < 1/4$, which we will show is the distribution resulting from applying algorithm fix to a sample of $\mathcal{G}_d(n)$.

First, we note that $\mathcal{G}_d$ has nice bicycle-freeness. We quote the relevant result from [Bor19], which we restate below:

Lemma 6.3.1. *([Bor19, Lemma 9]). Let $d \geqslant 3$ and r be positive integers. Then $G \sim \mathcal{G}_d(n)$ is bicycle-free at radius r with probability $1 - O((d-1)^{4r}/n)$.*

An obvious corollary of this is that for any constant $c < 1/4$, we have that $G \sim \mathcal{G}_d(n)$ is bicycle free at radius $c \log_{d-1} n$ with high probability.

To bound the number of short cycles in $\mathcal{G}_d(n)$ we use a classic result that very accurately estimates the number of short cycles in random regular graphs.

Lemma 6.3.2. *([MWW04, Section 2]). Let $G \sim \mathcal{G}_d(n)$ and X_i be the random variable that denotes the number of cycles of length i in G. Let $R_i = \max\{(d-1)^i/i, \log n\}$. Then*

$$\textbf{\textit{Pr}}\left[X_i \leqslant R_i, \textit{ for all } 3 \leqslant i \leqslant 1/4 \log_{d-1} n\right] = 1 - o_n(1).$$

Given the above, we obtain the following bound, for all $c < 1/4$:

$$\sum_{i=1}^{c \log_{d-1} n} \max\{(d-1)^i/i, \log n\} = O(n^c).$$

So we obtain the following proposition:

Proposition 6.3.3. *For any $c < 1/4$ and any $\epsilon > 0$, $G \sim \mathcal{G}_d(n)$ is a $(c \log_{d-1} n, O(n^c))$-graph and satisfies $\lambda(G) \leqslant 2\sqrt{d-1} + \epsilon$ with probability $1 - o_n(1)$.*

Finally, we want to apply Theorem 6.1.2, so first we need to verify its preconditions. For all $c < 1/4$ we have that $(2/3) \log_{d-1}(n/n^c) = (2/3)(1-c) \log_{d-1} n \geqslant c \log_{d-1} n$. Also note that

$n^c(d-1)^{c/2\log_{d-1} n+1} = n^{3c/2} = O(n^{3/8})$, so when applying Theorem 6.1.2 the resulting graph has $n + O(n^{3/8}) = n(1 + o_n(1))$ vertices. Thus, we obtain Theorem 6.1.4.

Remark 6.3.4. Recall that $\mathcal{G}_d(n)$ is the same as the conditional distribution of the d-regular n-vertex configuration model when conditioned on it being a simple graph. Indeed, a graph drawn from the d-regular n-vertex configuration model is simple with probability $\Omega_d(1)$. A result very similar to Lemma 6.3.2 also holds for the configuration model and thus the results of this section also hold for the configuration model.

6.3.1 Counting near-Ramanujan graphs with high girth

We will briefly prove Corollary 6.1.5 using the result we just proved. For simplicity, we are going to work with the configuration model, using the observation of Remark 6.3.4.

Our proof will use a classic result on the number of not necessarily simple d-regular n-vertex graphs, which is the same as the number of graphs in the n-vertex d-regular configuration model. It is easy to show [BC78] that for nd even, the number of such graphs is

$$\sim \left(\left(\frac{d^d n^d}{e^d (d!)^2} \right)^{n/2} \right).$$

Hence, the core claim we need to prove, is the following:

Proposition 6.3.5. *Let G_1 and G_2 be distinct graphs that follow the preconditions of Theorem 6.1.2. Then $\mathrm{fix}(G_1)$ and $\mathrm{fix}(G_2)$ are also distinct.*

This proposition implies that given any two good d-regular n-vertex graphs, applying fix produces two distinct graphs. From our proof of Theorem 6.1.4 we also know that the result of applying fix adds at most $O(n^{3/8})$ vertices. Finally, since a $(1 - o_n(1))$ fraction of the graphs are good an thus when we apply fix they result in $(2\sqrt{d-1} + \epsilon, c\log_{d-1} n)$-good graphs, the result follows. For briefness, we will not give a detailed proof but only a sketch of the proof.

Proof sketch of Proposition 6.3.5. Recall the H graph from the description of fix and let H_1 be such graph corresponding to G_1 and define H_2 analogously. If H_1 and H_2 are distinct, then $\mathrm{fix}(G_1)$ and $\mathrm{fix}(G_2)$ are also distinct. This follows from the fact that the vertices of the two added trees will have to be matched up in an isomorphism between $\mathrm{fix}(G_1)$ and $\mathrm{fix}(G_2)$.

We claim that if G_1 and G_2 are distinct, then H_1 and H_2 are distinct. Let S_i be the set of vertices that were endpoints of edges removed from cycles in G_1 and G_2, respectively. Note that there are at least two such vertices in S_i and also we cannot remove multiple edges adjacent to one vertex since this would imply the existence of two cycles in a small neighborhood, breaking the bicycle-freeness assumption. We can ignore the other removed edges since the local neighborhoods of edges removed from cycles are necessarily distinct from the local neighborhoods of the other removed edges. Now, the edges removed from G_i form a perfect matching on S_i that adds exactly $|S_i|/2$ cycles to H_i. Also, there is exactly one perfect matching that adds $|S_i|/2$ cycles to H_i to recover G_i. That means that there is only one G_i that could have generated H_i, which implies the claim. □

6.4 Explicit near-Ramanujan graphs of girth $\Omega(\sqrt{\log n})$

In this section we prove Theorem 6.1.6, building on the construction in the proof of Chapter 4. We note that the original construction has no guarantees on the girth of the constructed graph other than a constant girth.

To prove Theorem 6.1.6 we apply a similar strategy as the one from Section 6.3. Instead of derandomizing Lemma 6.3.2 we are going to obtain a simpler bound, which is good enough to obtain the desired. We note however, that Lemma 6.3.2 can be derandomized and for completeness we show how to in Section 6.4.1.

We start by proving the following lemma:

Lemma 6.4.1. *Let G be a d-regular n-vertex graph with $\lambda(G) \geqslant 2\sqrt{d-1}$ and such that G is bicycle-free at radius $\alpha \log_{d-1} n$, for $\alpha \leqslant 2$. Then we can apply fix to G and obtain a graph such that:*

- $\mathrm{fix}(G)$ *is d-regular and has $n(1+o_n(1))$ vertices;*
- $\lambda(\mathrm{fix}(G)) \leqslant \lambda(G) + o_n(1)$*;*
- $\mathrm{fix}(G)$ *has girth $(\alpha/3)\log_{d-1} n$.*

Before proving this lemma, we prove a core proposition in a slightly more generic way.

Proposition 6.4.2. *Let G be a d-regular graph that is bicycle-free at radius $2r$, then*

$$|\mathrm{Cyc}_r(G)| \leqslant n/(d-1)^r.$$

Proof. Pick one vertex per cycle in $\mathrm{Cyc}_r(G)$ and place it in a set S. We claim that for every distinct $u, v \in S$, $B_r(u) \cap B_r(v) = \emptyset$. Suppose this wasn't the case and suppose there is some w such that $w \in B_r(u) \cap B_r(v)$, for some pair u, v. Note that $B_{2r}(w)$ includes the two length r cycles that correspond to u and v, which contradicts bicycle-freeness in G.

Given the above, we have that the sets $B_r(u)$ for $u \in S$ are pairwise disjoint and also we know that $|B_r(u)| = d(d-1)^{r-1}$. Hence we have:

$$|\mathrm{Cyc}_r(G)| \cdot d(d-1)^{r-1} \leqslant n,$$

which implies the desired result. □

And we can prove the above lemma.

Proof of Lemma 6.4.1. By plugging G into Proposition 6.4.2 we can conclude that G is a $(\alpha \log_{d-1} n, n^{1-\alpha/2})$-graph. We wish to apply Theorem 6.1.2 so first recall its preconditions. By definition $\lambda(G) \geqslant 2\sqrt{d-1}$. However, the precondition on the radius of bicycle-freeness does not hold, since $(2/3)\log_{d-1}(n/n^{1-\alpha/2}) = (\alpha/3)\log_{d-1} n$ which is less than $\alpha \log_{d-1} n$. If we instead use the fact that G is also trivially a $((\alpha/3)\log_{d-1} n, n^{1-\alpha/2})$-graph, then the precondition is satisfied.

Thus, we can apply Theorem 6.1.2 and we obtain that $\mathrm{fix}(G)$ satisfies all the required conditions, which concludes the proof. □

Given this lemma, we will modify the first step of the construction of Chapter 4 to produce a graph G_0 with girth $c\sqrt{\log n}$. Note that, similarly to bicycle-freeness, the girth of a graph

can only increase when applying any 2-lift, so this strategy guarantees that after step 2 of the construction, the final graph has the desired girth, which would imply Theorem 6.1.6.

First, when enumerating over all seeds to generate G_0 in step 1, we look for one that guarantees that G_0 is bicycle-free at radius $(1/5)\log_{d-1} n_0$ (recall that by Theorem 1.1.16 a $1 - o_n(1)$ fraction of the seeds satisfy this). Next, we apply Lemma 6.4.1 and obtain $\mathrm{fix}(G_0)$ with girth $(1/15)\log_{d-1} n_0$ and the desired value of $\lambda(G_0)$. Let $\kappa = 15c/\log_{d-1} 2$. We can set n_0 to $2^{\kappa\sqrt{\log n}}$, in which case G_0 has girth $c\sqrt{\log n}$.

Note that the above only works as long as $\kappa \leqslant \sqrt{\log n}$, otherwise $n_0 > n$. Also, from Theorem 4.1.1 and Theorem 1.1.16, we need $d \leqslant (\log n)^{1/8}/C$ and $\epsilon \gg \sqrt{d}(\log\log n)^4/(\log n)$ (the details on how to obtain these can be found on Chapter 4).

Finally, we can precisely determine the running time of this algorithm. From Theorem 1.1.16, constructing G_0 takes time $\mathrm{poly}(n_0^{\log(n_0)/\sqrt{\epsilon}}) = \mathrm{poly}(n^{\log(c/\log_{d-1}(2))/\sqrt{\epsilon}})$ and using Theorem 4.1.1 with the appropriate choice of C takes time $\mathrm{poly}(n^{d^{1/4}\log(d)/\sqrt{\epsilon}})$.

6.4.1 Derandomizing the number of short cycles

Proposition 6.4.3. *Fix $d \geqslant 3$, n and $k \geqslant c\log_{d-1} n$, where $c < 1/4$. Let G be drawn from the d-regular n-vertex $4k$-wise configuration model and X_i be the random variable that denotes the number of cycles of length i in G. Let $R_i = \max\{(d-1)^i/i, \log n\}$. Then*

$$\textbf{\textit{Pr}}\left[X_i \leqslant R_i, \textit{ for all } 1 \leqslant i \leqslant 1/4\log_{d-1} n\right] = 1 - o_n(1).$$

By Theorem 3.4.5, these statements remain true in the $(\delta, 4k)$-wise uniform versions of the model, $\delta \leqslant 1/n^{16k+1}$.

Proof. The proof follows almost directly from the proof of Lemma 6.3.2. First, note that X_i can be written as a polynomial of degree at most i in the entries of G's adjacency matrix, by summing over the products of the edge indicators of all possible cycles of length i in G. Thus, it can be written as a polynomial of degree at most $2k$ in the permutation indicators $1[\pi(j) = (v, i)]$. So we can compute $\mathbf{E}[X_i]$ assuming that X_i is drawn from the fully uniform configuration model. Similarly, X_i^2 can be written as a polynomial of degree at most $4k$ in the permutation indicators, so we can compute $\mathbf{Var}[X_i]$ assuming that X_i is drawn from the fully uniform configuration model.

From [MWW04] we have the following estimates, that only apply when $(d-1)^{2i-1} = o(n)$:

$$\mathbf{E}[X_i] = \frac{(d-1)^i}{2i}(1 + O(i(i+d)/n)) \qquad \mathbf{Var}[X_i] = \mathbf{E}[X_i] + O(i(i+d)/n)\mathbf{E}[X_i]^2.$$

By applying Chebyshev's inequality to each X_i, just like in [MWW04], we get the desired result. □

We can finally rewrite Theorem 1.1.16 in the language of the d-regular n-vertex (δ, k)-wise uniform configuration model and tack on the result we just proved.

Theorem 6.4.4. *For a large enough universal constant α and any integer $n > 0$, fix $3 \leqslant d \leqslant \alpha^{-1}\sqrt{\log n}$ and $c < 1/4$, and let $\varepsilon \leqslant 1$ and k satisfy*

$$\varepsilon \geqslant \alpha^3 \cdot \left(\frac{\log\log n}{\log_{d-1} n}\right)^2, \qquad k \geqslant \alpha \log(n)/\sqrt{\varepsilon}.$$

Let G be chosen from the d-regular n-vertex k-wise uniform configuration model. Then except with probability at most $1/n^{.99}$, the following hold:

- *G is bicycle-free at radius $c \log_{d-1} n$;*
- *The total number of cycles of length at most $c \log_{d-1} n$ is $O(n^c)$;*
- *$\lambda(G) \leqslant 2\sqrt{d-1} \cdot (1+\varepsilon)$.*

Finally, by Theorem 3.4.5, these statements remains true in the (δ, k)-wise uniform configuration model, $\delta \leqslant 1/n^{16k+1}$.

Chapter 7

Abelian Lifts and Applications to Coding Theory

In this section we describe the results of [JMO+22], which was joint work with Fernando Granha Jeronimo, Tushant Mittal, Ryan O'Donnell and Madhur Tulsiani. In this paper we study a generalization of lifts based on groups and we describe explicit constructions of expanders obtained via abelian lifts. Expanding graphs obtained via abelian lifts, form a key ingredient in recent breakthrough constructions of quantum LDPC codes. However, these constructions are non-explicit. Our result obtains explicit quantum lifted product codes of almost linear distance (and also in a wide range of parameters) based on these non-explicit results. As a corollary, we also obtain good quasi-cyclic LDPC codes with any circulant size up to nearly linear.

7.1 Symmetries and codes

Regarding explicit constructions of expanding graphs, one of the goals of this thesis so far has been to construct graphs that have both strong spectral bounds and some other property. In this chapter our focus is on constructing graphs that have certain *symmetries*. Informally, we say that G has symmetries of H if $H \subseteq \mathrm{Aut}(G)$, where $\mathrm{Aut}(G)$ denotes the group of all graph isomorphisms to itself.

One of the problems that has been studied in graph theory is to construct graphs with a given automorphism group. Frucht [Fru39] proved in 1939 that for every finite group H, we have a graph G such that $\mathrm{Aut}(G) = H$. Later, Babai [Bab74] showed that there is such a graph on at most $2|H|$ vertices, except for $\mathbb{Z}_3$, $\mathbb{Z}_4$ and $\mathbb{Z}_5$. Thus, one can ask if there are explicit constructions of expanding graphs with given symmetries.

While interesting in its own right, the ability to control symmetries also has concrete applications. For example, a very recent work [GW21] constructs many families of expanding asymmetric graphs, i.e., having no symmetries, and shows applications to property testing and other areas. We will focus on an important connection to both quantum and classical codes that was the motivation behind this work.

Low-density parity check (LDPC) codes were first introduced by Gallager [Gal62] in the '60s and are one of the most popular classes of classical error-correcting codes, both in theory

and in practice. LDPC codes are linear codes whose parity check matrices have row and column weights bounded by a constant (which means that each parity check depends only on a constant number of bits). The popularity of this family of codes comes from the fact that there are many known constructions of classical LDPC codes that achieve linear rate and distance that can also be decoded in linear time [RU08].

A family of codes that has been extensively studied is *cyclic codes*. These are codes that are invariant under the action of $\mathbb{Z}_N$ where N is the block length. This symmetry leads to efficient encoding and decoding algorithms and a major open problem is whether good cyclic codes exist. Babai, Shpilka and Stefankovich [BSS05] showed that cyclic codes cannot be good LDPC codes and this negative result was extended by Kaufman and Wigderson [KW10] to LDPC codes with a *transitive* action by an arbitrary abelian group.

Quasi-cyclic codes are a generalization of cyclic codes in which symmetry is only under rotations of multiples of a parameter (called index) n where $N = n\ell$. This is equivalent to relaxing the transitivity condition to allow for n orbits. Unlike cyclic codes, good quasi-cyclic codes are known to exist as was shown by Chen, Peterson and Weldon [CPW69]. More recently, Bazzi and Mitter [BM06] gave a randomized construction for any constant $n > 2$ and showed that it attains Gilbert–Varshamov bound rate $1/n$. Quasi-cyclic codes have been extensively studied and are very useful in practice (e.g., their LDPC counterparts are part of the 5G standard of mobile communication [LBM$^+$18]).

In the realm of quantum computing, the fragility of qubits makes quantum error correcting codes crucial for the realization of scalable quantum computation. *Calderbank-Shor-Steane* (CSS) codes are a family of quantum error-correcting codes that was first described in [CS96, Ste96]. A CSS code is defined by a pair of classical linear codes that satisfy an orthogonality condition. The quantum analog of LDPC codes is thus defined as CSS codes where the parity check matrices of both codes have bounded row and column weights.

Constructing quantum LDPC codes of large distance has been active area of research recently. After two decades, [EKZ20] broke the $\sqrt{N}$ barrier and there was a flurry of activity with [HHO21] extending it to $N^{3/5}$ (up to $\operatorname{poly}\log$ factors). Panteleev and Kalachev [PK21] came up with another breakthrough construction achieving almost linear distance. Both [HHO21] and [PK21] are non-explicit constructions crucially relying on symmetries. The construction in [PK21] interestingly used quasi-cyclic LDPC codes which in turn was constructed using expander graphs with cyclic symmetry. Moreover, Breuckmann and Eberhardt [BE21] introduced a new approach for constructing quantum codes simultaneously generalizing [HHO21] and [PK21] in order to obtain explicit codes out of a pair of graphs having the symmetries of any group. This provides a very concrete motive to study explicit construction of expander graphs symmetric under various families of groups.

Recently Panteleev and Kalachev [PK22] showed how to construct explicit asymptotically good quantum LDPC codes, which subsumed our application of the results of this chapter to quantum LDPC codes.

Current techniques

Many of the current known constructions of expanders are Cayley graphs and therefore are highly symmetric but are somewhat rigid in the sense that one may not be able to finely control the

symmetries of a given construction. One general approach is to construct an expanding Cayley graph for a given group but the Alon–Roichman theorem [AR94] only guarantees a logarithmic degree which is tight when the group is abelian (and this large degree is undesirable for some application in coding theory). The other technique used to build expanders is via lifting, like seen in the previous chapters.

Let's consider the graph lift operation. One way to generalize a graph lift is by restricting the types of matchings that are allowed to replace an edge. In general form, a group lifting operation takes a lift size parameter ℓ, a base graph G_0 on n vertices and a subgroup H of the symmetric group $\mathrm{Sym}(\ell)$ and constructs a new "lifted" graph G on $n\ell$ vertices where each vertex v of G_0 is replaced by ℓ-copies $(v,1),\ldots,(v,\ell)$ and for every edge $e=(u,v)$ of G_0 we associate an element of $h_e \in H$ and (u,i) is connected to $(v,h_e(i))$ for $i \in [\ell]$. Notice that an ordinary ℓ-lift, which we will call a *unstructured* ℓ-lift for the rest of this chapter, is a lift based on the symmetric group, i.e. $H=\mathrm{Sym}(\ell)$.

An example of such group is the class of *shift lifts*, where we consider the cyclic group $\mathbb{Z}_\ell$. These were first studied by Agarwal et al. [ACKM19], who showed that an uniformly random shift k-lift of any n-vertex d-regular base graph G has the new eigenvalues bounded by $\lambda(G)+O(\sqrt{d})$ with probability $1-k\cdot\exp(-\Omega(n/d^2))$. Later, Chandrasekaran and Velingker [CV17] showed that for bipartite graphs there is always a shift 3-lift and a 4-lift whose new eigenvalues are bounded by $2\sqrt{d-1}$, using techniques similar to those of [MSS15a]. They also conjectured that this is true for any shift k-lift, but Agarwal et al. [ACKM19] showed that for $k=2^{\Omega(nd)}$ this is impossible (and the same applies to any lift based on an abelian group).

Lifting has three very useful properties. One, it preserves the degree of the base graph. Secondly, random lifts preserve expansion with high probability. This holds for any lift size in the case of "unstructured" ℓ-lifts, but only holds for $\ell \leqslant 2^{O_d(n)}$ when H is abelian (and transitive). Finally, if H is abelian, then the lifted graph inherits symmetries of H. The first two properties are clearly useful in constructing larger expanders from a small one, and for this reason, there has been extensive work on lift based constructions.

Motivated by the applications of these lifts to codes, we obtain explicit constructions of expanding abelian lifts, for a wide range of lift sizes.

7.1.1 Our results and techniques

Consider an n-vertex d-regular graph $G=(V,E)$ and assume that we have an an ordering on V and by convention, $(u,v)\in E$ if $u \leqslant v$.

The action of a group H on a set of ℓ elements is defined by a map $\psi : H \to \mathrm{Sym}(\ell)$ which satisfies $\psi(h_1h_2)=\psi(h_1)\psi(h_2)$. Since we only care about the action of the group, we will assume that our input is actually $\psi(H)\subseteq\mathrm{Sym}(\ell)$ and the action is the natural one.

Definition 7.1.1 ((H,ℓ)-lift of a graph)**.** An (H,ℓ)-signing of an undirected graph $G=(V,E)$ is a function $s : E \to H \subseteq \mathrm{Sym}(\ell)$. The lifted graph $G(s)=(V',E')$ is a graph on ℓ copies of the vertices $V'=V\times[\ell]$ where for every edge $(u,v)\in E$ we have $((u,i),(v,s(u,v)\cdot i))\in E'$

We will restrict to analyzing abelian H and the most important case to consider is when $H=\mathbb{Z}_\ell$, i.e. the cyclic group. A necessary condition for the lift to be expanding is for it to be connected. A subgroup H is *transitive* if for every $i,j\in[\ell]$, there exists $h\in H$ such that

$h \cdot i = j$. Lifts of non-transitive subgroups are disconnected because if the pair $\{i, j\}$ violate the condition then any pair (u, i) and (v, j) are disconnected. Thus, we will assume henceforth that we work with transitive abelian subgroups.

Our construction of the lifts (and the expansion thereof) vary based on the parameter ℓ and we make the following classification for ease in presenting the results. Let n, d, ε be given and suppose we want an n-vertex d-regular graph with spectral expansion a function of ε.

- *Sub-Exponential* - This is the regime where $\ell \leqslant \exp\left(n^{\delta(d,\varepsilon)}\right)$. The exponent $\delta(d, \varepsilon)$ goes to zero as the degree (d) increases or ε vanishes.
- *Moderately-Exponential* - This is when $\ell \leqslant \exp\left(n^{\delta_0}\right)$. The exponent is some fixed universal constant $\delta_0 \in (0, 1)$.

Our first main result shows explicit constructions in the sub-exponential and moderately exponential regimes.

Theorem 7.1.2. *For large enough n and constant degree $d \geqslant 3$, given ℓ such that $\ell \leqslant \exp(n^{\Theta(1)})$, the generating elements of a transitive abelian group $H \leqslant \mathrm{Sym}(\ell)$, and any fixed constant $\varepsilon \in (0, 1)$, we can construct in deterministic polynomial time, a d-regular graph G on $\Theta(n\ell)$ vertices such that*

- *G is an (H, ℓ)-lift of a graph G_0 on $\Theta(n)$ vertices.*
- *(Sub-Exponential) If $\ell \leqslant \exp\left(n^{\delta(d,\varepsilon)}\right)$, then $\lambda(G) \leqslant 2\sqrt{d-1} + \epsilon$.*
- *(Moderately-Exponential) If $\ell \leqslant \exp\left(n^{\delta}\right)$ and also $d \geqslant d_0(\epsilon)$, then $\lambda(G) \leqslant \varepsilon \cdot d$.*

The bulk of the technical work is in the proof of Theorem 7.1.2. For this, we build on the techniques described in Chapter 4 for derandomizing 2-lifts via the trace power method. When analyzing larger lift sizes (required in our derandomization of quantum and classical codes), we are led to consider much larger walk lengths in the trace power method. For lift sizes larger than $2^{2^{\Theta(\sqrt{\log n})}}$, the counting argument in Chapter 4 trivializes no longer implying expansion of the construction. Our main technical contribution consists in providing alternative ways of counting such special walks by carefully compressing the traversal of the depth-first search (DFS) algorithm.

We are able to extend the near-Ramanujan guarantee for 2-lifts from Chapter 4 to the entire sub-exponential regime of lift sizes ℓ. In the moderately exponential regime, the walks are too long and we resort to another counting that can only guarantee an expansion of $\varepsilon \cdot d$. Theorem 7.1.2 can be seen as a simplification of the construction from Chapter 4 since we can now do a single large lift instead of performing a sequence of 2-lifts.

Let us now formally state the results of Agarwal et al. in Theorem 7.1.3 showing *randomized* constructions of abelian lifts.

Theorem 7.1.3 (Agarwal et al. [ACKM19], Theorem 1.2)**.** *Let G_0 be a d-regular n-vertex graph, where $2 \leqslant d \leqslant \sqrt{n/(3 \ln n)}$. Let G be a random $(\mathbb{Z}_\ell, \ell)$-lift of G_0. Then*

$$\lambda(G) = O(\lambda(G_0)),$$

with probability $1 - \ell \cdot e^{-\Omega(n/d^2)}$. Moreover, if $\ell \geqslant \exp(O_\varepsilon(nd))$, then no abelian (H, ℓ) lift has $\lambda(G) \leqslant \varepsilon \cdot d$.

This result is based on discrepancy methods building on the work of Bilu and Linial [BL06] and gives lower and upper bounds that are tight up to a factor of d^3 in the exponent.

Theorem 7.1.2 can be seen as a (derandomization of the parameters) in Theorem 7.1.3 for every constant degree and lift size from 2 all the way to $\exp(n^{\Theta_d(1)})$. In the sub-exponential regime, our result improves their spectral guarantee from $O(\sqrt{d})$ to $2\sqrt{d-1}+\epsilon$.

7.1.2 Derandomized quantum and classical codes

We first state the code constructions in [PK21] and then show how large explicit abelian lifts derandomize their codes.

Theorem 7.1.4 ([PK21]). *Let G be a d-regular graph on $n\ell$-vertices such that G has a symmetry*[1] *of $\mathbb{Z}_\ell$ and $\lambda_2(G) \leqslant \varepsilon \cdot d$. Then we can construct the following,*

- *A good quasi-cyclic LDPC code of block length $N = \Theta(n\ell)$ and index $\Theta(n)$.*
- *A quantum LDPC code which has distance $\Theta_{\epsilon,d}(\ell)$ and dimension $\Theta(n)$.*

Panteleev and Kalachev use the aforementioned *randomized* construction of abelian lifted expanders by Agarwal et al. [ACKM19], where each edge of the base graph is associated with an element in $\mathbb{Z}_\ell$ sampled uniformly. When ℓ is in the exponential regime they obtain quantum LDPC codes with almost linear distance, i.e., $\Omega(N/\log(N))$.

Breuckmann and Eberhardt [BE21] gave a derandomization of [PK21] in a more restricted parameter regime by observing that the Ramanujan graph construction by Lubotsky, Philips and Sarnak [LPS88] of size n has a (free) action of $\mathbb{Z}_{n^{1/3}}$. By Theorem 7.1.4, we have an explicit quantum LDPC code of distance $O(N^{1/3})$ under the notion of distance[2] in [PK21, HHO21].

As a direct corollary of Theorem 7.1.2, we have a partial derandomization of [PK21] yielding explicit quantum LDPC codes of almost linear distance. This greatly improves the distance of the existing explicit construction. We also get good quasi-cyclic LDPC codes of almost linear circulant size. Moreover, the ability to construct a wide range of lift sizes from Theorem 7.1.2 lets us control the circulant size which can be useful in practice. By controlling the lift size, we can also directly amplify the rate of their quantum LDPC codes (without resorting to the product of complexes). To summarize,

Corollary 7.1.5 ([PK21], Theorem 7.1.2). *We have an explicit polynomial time construction of each of the following,*

- *Good quasi-cyclic LDPC code of block length N and any circulant size up to $N/\mathrm{polylog}(N)$.*
- *Quantum LDPC code with distance $\Omega(N/\mathrm{polylog}(N))$ and dimension $\Omega(\mathrm{polylog}(N))$.*
- *Quantum LDPC code with distance $\Omega(N^{1-\alpha/2}/\mathrm{polylog}(N))$ and dimension $\Theta(N^\alpha \mathrm{polylog}(N))$ for every constant $\alpha > 0$.*

7.2 Non-backtracking walks and the Ihara–Bass formula for group lifts

Like was done on Chapter 5, we will need new tools to deal with non-backtracking walks in the abelian lift model. We will see that we will need to use representation theory of abelian groups

[1] To be more precise, $\mathbb{Z}_\ell$ acts freely on G.

[2] [BE21] state their result for a slightly different notion of a quantum codes called subsystems codes for which the corresponding distance (also known as dressed distance) is larger.

to facilitate this discussion so we start by defining some well known concepts.

A *character*[3] of a group is a map $\chi : H \to \mathbb{C}^*$ that respects group multiplication, i.e., $\chi(h_1h_2) = \chi(h_1)\chi(h_2)$. For a finite group $|\chi(h)| = 1$ for every $h \in H$. The *trivial character* is the one which has $\chi(h) = 1$ for every h. The rest of the characters we call *non-trivial*.

Recall that an (H, ℓ)-signing of an undirected graph $G = (V, E)$ is a function $s : E \to H \subseteq \mathrm{Sym}(\ell)$. We extend the signing to $\vec{E}$ such that for an edge $(u, v) \in E$, $s(v, u) := s(u, v)^{-1}$.

Definition 7.2.1 (Non-backtracking walk operator)**.** For an extended signing $s : \vec{E} \to H$ and a character χ of H, the signed non-backtracking walk matrix $B_s(\chi)$ is a non-symmetric matrix of size $|\vec{E}| \times |\vec{E}|$ in which the entry corresponding to the pair of edges $(u, v), (x, y)$ is $\chi(s(x, y))$ if $v = x,\ u \neq y$, and zero otherwise.

The unsigned variant is obtained by taking the trivial character in the definition above. Let the non-backtracking walk matrix of G be B and the lifted graph with respect to a signing s be $B_{G(s)}$.

For the rest of this section we will prove two important facts that we summarize here:

Fact 7.2.2. *Let B be the non-backtracking walk matrix of a d-regular graph G. Then,*

$$\lambda(G) \leqslant 2 \cdot \max\{\sqrt{d-1},\ \rho_2(B)\}.$$

Fact 7.2.3. *If $H \subseteq \mathrm{Sym}(\ell)$ is abelian, then there exist characters $\{\chi_1, \cdots, \chi_\ell\}$[4] such that we have $\mathrm{Spec}(B_{G(s)}) = \bigcup_i \mathrm{Spec}(B_s(\chi_i))$. If H is transitive, then exactly one of the characters is trivial.*

7.2.1 Diagonalizing the non-backtracking operator

Let $\rho : Sym(l) \to \mathrm{GL}(\mathbb{C}^l)$ be the matrix representation of a permutation. More concretely, given a permutation $\sigma \in \mathrm{Sym}(l)$ the map $\rho(\sigma)e_i = e_{\sigma \cdot i}$ where $\{e_1, \cdots, e_l\}$ is the set of elementary basis vectors of $V = \mathbb{C}^l$. Since $H \subseteq \mathrm{Sym}(l) \to \mathrm{GL}(\mathbb{C}^l)$ it also gives a map on H by restriction. For example, let P be the $l \times l$ permutation matrix that maps $Pe_i = e_{i+1}$ where $i + 1$ is taken modulo l. Then for $H = \mathbb{Z}_l$ and for $t \in \mathbb{Z}_l$, $\rho(t) = P^t$.

For a map ρ as above and an extended signing s , define a generalized non-backtracking walk matrix in which for a non-zero entry indexed by (e_1, e_2) we replace 1 by the block matrix $\rho(s(e_2))$.

Lemma 7.2.4. *The non-backtracking walk matrix of the lifted graph is $B_{G(s)} = B_G(\rho)$.*

Proof. In the lifted graph, the edges are of the form $[(u, i - s(u, v)), (v, i)] =: [u, v, i]$ and thus can be indexed by $E' \times [l]$. The non-backtracking walk matrix $B_{\hat{G}}$ would then have a non zero entry from $([u, v, i], [x, y, j])$ iff $(v, i) = (x, j - s(x, y))$ and $(y, j) \neq (u, i - s(u, v))$. Assume that the first condition is met i.e. $x = v$ and $j = i + s(x, y)$. If $y = u$, then $i - s(u, v) = i - s(y, x) = i + s(x, y) = j$ and therefore, the second condition can't be met. This is just a longer way of saying that the lifts give a matching between $u \times [l]$ and $v \times [l]$. The implication of all this is that y has to be distinct from u and thus the pair of edges $(u, v), (v, y)$ has a non-zero

[3]The definition we give is that of a *linear character*. We use the term character as we work only with abelian groups.

[4]These need not be distinct. For example if H is trivial, then all the χ_i are trivial

entry in B_G. Moreover, for every i and every pair of edges $(u,v),(v,y)$ we have a non-zero entry for $(u,v,i),(v,y,i+s(v,y))$ in $B_{G(s)}$ and thus it can be written as a block matrix with the entry in $(u,v),(v,y)$ equal to $\rho(s(v,y))$. □

Since the base graph G and the signing s will be fixed throughout, we will drop the subscript to make reading less hurtful. We will need the following well-known fact (see [Con, Thm. 5] for a proof) that a collection of commuting matrices that are diagonalizable are also simultaneously diagonalizable. Since, H is abelian, we have that $\{\rho(h)\}$ are commuting and since they are invertible, they clearly are diagonalizable. Thus, they simultaneously diagonalize, i.e., there exists F such that $\rho(h) = F\,\mathrm{diag}(\chi_1(h),\cdots,\chi_l(h))F^{-1}$ where χ_i are characters of H.

Corollary 7.2.5. *If for H, the standard representation splits as $\rho = \oplus_i \chi_i$, then the non-backtracking walk matrix $B_G = Q\,\mathrm{diag}(B(\chi_1),\cdots,B(\chi_t))Q^{-1}$ and thus $\mathrm{Spec}(B_G) = \cup_i \mathrm{Spec}(B(\chi_i))$. Moreover, if H is transitive, then exactly one of the characters is trivial.*

Proof. To ease notation we write $B_G(\rho) = \sum M_{u,v} \otimes \rho(s(u,v))$ for some $M_{u,v}$. We have $\rho(s(u,v)) = F\,\mathrm{diag}(\chi_1(h),\cdots,\chi_l(h))F^{-1}$ and thus

$$B_G(\rho) = (I \otimes F)\sum M_{u,v} \otimes \mathrm{diag}(\chi_1(h),\cdots,\chi_l(h))(I \otimes F^{-1})$$

Let $|E| = N$ and let T denote the permutation on Nt that maps $T(i) := bt + (a+1)$ where a, b are the unique non-negative integers such that $0 \leqslant b < N$ $i - 1 = aN + b$. It can then be seen that $\sum M_{u,v} \otimes \mathrm{diag}(\chi_1(h),\cdots,\chi_t(h)) = T\,\mathrm{diag}\left(\sum M_{u,v} \otimes \chi_i(h)\right) T^{-1}$. Notice that $\sum M_{u,v} \otimes \chi_i(h) = B_G(\chi_i)$ and thus putting it together we have that for $Q = (I \otimes F)T$, $B_{G(s)} = Q\,\mathrm{diag}(B(\chi_1),\cdots,B(\chi_t))Q^{-1}$. The statement on the spectrum follows immediately.

Since, the all-ones vector is clearly invariant under the standard representation, we have a copy of the trivial character χ_0 in ρ. Let there be another vector $v \in \mathbb{C}^\ell$ that is invariant. Let $i \in \mathrm{supp}(v)$ and $j \notin \mathrm{supp}(v)$. By transitivity, we have an h such that $h \cdot i = j$ but then $h \cdot v \neq v$ which violates the invariance. □

7.2.2 An Ihara–Bass formula for signed graphs

Claim 7.2.6. Let A be the (signed) adjacency matrix of a d-regular graph. Suppose f is an eigenvector of A satisfying

$$Af = \left(\beta + \frac{d-1}{\beta}\right) f.$$

Then $g(u,v) \coloneqq (f(u) - \beta f(v))$ (or in the signed case $g(u,v) \coloneqq A(u,v)^{-1}(f(u) - \beta \cdot A(u,v)f(v))$) is an eigenvector of the (signed) non-backtracking matrix B with eigenvalue β.

Proof. Let f and g be as in the statement of the claim. Suppose for that A and B are not signed. Computing we have

$$(Bg)(u,v) = \sum_{w \sim v, w \neq u} f(v) - \beta \cdot f(w)$$

$$
\begin{aligned}
&= (d-1)f(v) - \sum_{w\sim v, w\neq u} \beta \cdot f(w) \\
&= (d-1)f(v) + \beta \cdot f(u) - \beta \sum_{w\sim v} f(w) \\
&= (d-1)f(v) + \beta \cdot f(u) - \beta (Af)(v) \\
&= (d-1)f(v) + \beta \cdot f(u) - \beta \left(\beta + \frac{d-1}{\beta}\right) f(v) \\
&= \beta(f(u) - \beta \cdot f(v)) = \beta \cdot g(u,v).
\end{aligned}
$$

Now suppose that A and B are signed. First note that g is well-defined since for every entry $g(u,v)$ the pair (u,v) is an orientation of an edge of the graph so it has a signing $A(u,v) \neq 0$. We have

$$
\begin{aligned}
(Bg)(u,v) &= \sum_{w\sim v, w\neq u} A(v,w)A(v,w)^{-1}(f(v) - \beta \cdot A(v,w)f(w)) \\
&= (d-1)f(v) - \beta \sum_{w\sim v, w\neq u} A(v,w)f(w) \\
&= (d-1)f(v) + \beta \cdot A(v,u)f(u) - \beta \sum_{w\sim v} A(v,w)f(w) \\
&= (d-1)f(v) + \beta \cdot A(v,u)f(u) - \beta (Af)(v) \\
&= (d-1)f(v) + \beta \cdot A(v,u)f(u) - \beta \left(\beta + \frac{d-1}{\beta}\right) f(v) \\
&= \beta \cdot A(v,u) \left(f(u) - \beta \frac{1}{A(v,u)} f(v)\right), \\
&= \beta \cdot A(u,v)^{-1} \left(f(u) - \beta \cdot A(u,v)f(v)\right) = \beta \cdot g(u,v),
\end{aligned}
$$

where we used $A(v,u) = A(u,v)^{-1}$. □

Corollary 7.2.7. *Let A be the (signed) adjacency matrix of a d-regular graph. Let B be its (signed) non-backtracking operator. For any $\lambda > 2\sqrt{d-1}$, if $\rho_2(B) \leqslant \lambda/2$, then $\rho_2(A) \leqslant \lambda$. Hence, $\lambda(G) = \rho(A) \leqslant 2\max\{\sqrt{d-1}, \rho_2(B)\}$.*

Proof. We show via the contrapositive. Suppose that f is eigenvector of A with eigenvalue α such that $|\alpha| > \lambda$. By possibly multiplying A and B by a phase (i.e., $e^{i\theta}$), we can assume α is a non-negative real number. By Claim 7.2.6, we have that β satisfying the equation $\beta^2 - \alpha\beta + (d-1) = 0$ is an eigenvalue of B. Considering the solution $\beta^+ = (\alpha + \sqrt{\alpha^2 - 4(d-1)})/2$ and thus, we have $\beta^+ \geqslant \alpha/2 > \lambda/2$. □

7.3 Proof strategy

We give an overview of the proof of Theorem 7.1.2. To do so, we will first recall some of the techniques of Chapter 4. To avoid discussing some unimportant technicalities, we will make some simplifications in this high-level overview.

Let G_0 be a base expander graph and $s\colon E_0 \to \mathbb{Z}_2$ be a signing that defines a lift. As we have seen before, we can use the trace method to bound the spectral radius of the non-backtracking operator. Combining that

$$\rho(B_s)^{2k} \leqslant \operatorname{tr}((B_s^*)^k B_s^k) = \sum_{\substack{(e_1,\ldots,e_{2k}) \\ \text{closed edge walk}}} \prod_{i=1}^{2k} \chi(s(e_i)).$$

The above expression greatly simplifies when we take the expectation over a uniformly random signing since only walks in which every edge occurs at least twice stand a chance of surviving the expectation. We have

$$\mathbf{E}\left[\rho(B)^{2k}\right] \leqslant \sum_{\substack{(e_1,\ldots,e_{2k}) \\ \text{closed edge walk}}} \mathbf{E}_s\left[\prod_{i=1}^{2k} \chi(s(e_i))\right]$$

reducing the problem of bounding the spectral radius to a counting problem of these special walks. In the hypothetical scenario of G_0 being Ramanujan and the counting on the RHS above being $(d-1)^k$, we would have a Ramanujan lift. This idealized scenario can be too optimistic and the count of $(d-1)^k$ has additional factors, but they remain small after taking a $2k$-th root (when k is neither too small or large)

Now, we wish to count $2k$-length singleton free non-backtracking walks in G_0. For the sake of intuition, we will assume that G_0 has girth g, but has we saw before, we can modify the next argument to when the graph is merely bicycle-free at radius $g/2$. Assume that $g = \Omega(\log_{d-1}(n))$ and consider the hike graph $\mathcal{H}$. If k is not too large, then $\mathcal{H}$ looks like a tree possibly with a few additional edges forming cycles as established by Alon, Hoory and Linial in [AHL02].

Assuming that the hike is singleton free, we can have at most k steps that visit an edge that was not previously visited. This implies that the hike graph $\mathcal{H}$ has at most k edges and at most $k+1$ vertices (since it is connected). We can count the number of these special walks by directly specifying an encoding for the hike. Up to negligible factors (after $2k$-th root for k not too small), there are at most

$$n \cdot (d-1)^k \cdot k^{O\left(\frac{\ln(k)}{g}\right)\cdot k},$$

singleton free hikes of length $2k$. This bound trivializes, i.e., it becomes at least $(d-1)^{2k}$, for $\ln(k) \gg \sqrt{g} = \Theta\left(\sqrt{\log_{d-1}(n)}\right)$. This means that we cannot use this bound for very long walks and this in turn prevents us from getting lift sizes larger than $2^{2^{\Theta(\sqrt{\log_{d-1}(n)})}}$ from this result.

Now, let's turn to the setup of this Chapter. Consider $\mathbb{Z}_\ell$ lifts for large ℓ. The spectral radius of each individual $B_s(\chi)$ can be analyzed in a similar fashion as above via the trace power method. However, we need to bound all of them *simultaneously*. We know no better way than a simple union bound over the $\ell - 1$ cases, but this will force us to obtain a much better concentration guarantee out of the trace power method which in turn entails having to consider much larger walk lengths.

Instead of encoding a hike directly, we will first encode the subgraph of G_0 traversed by the hike (the hike graph) and then encode the hike having the full hike graph at our disposal. We will give two different encodings for the hike graph. The first one is simpler and can encode an arbitrary graph. The second encoding uses the special structure of the hike graph, namely, having few vertices of degree greater than 2. Both encodings are based on the traversal history of the simple depth-first search (DFS) algorithm. Let $\mathcal{H}$ be the hike graph on $m \leqslant k$ edges and $n' \leqslant k+1$ vertices. As DFS traverses $\mathcal{H}$, each of its edges will be visited twice: first "forward" via a recursive call and later "backwards" via a backtracking operation. We view each step of the DFS traversal as being associated with an edge that is being currently traversed and the associated type of traversal: recursive (R) or backtracking (B). A key observation is that only for the recursive traversals we need to know the next neighbor out of $d-1$ possibilities (except for the first step). For the backtracking steps, we can rely on the current stack of DFS. Thus, if we are given a starting vertex from G_0, a binary string in $\{ R, B \}^{2m}$ and a next neighbor for each recursive step, we can reconstruct $\mathcal{H}$. Note that there at most

$$n \cdot d \cdot (d-1)^k \cdot 2^{2k},$$

such encodings. Having access to the hike graph and again assuming that the graph has girth $g = \Omega(\log_{d-1}(n))$ (similarly, bicycle freeness is also enough). Using the locally tree-like structure, a $2k$-length hike can be specified by splitting it into segments of length $< g/2$, by specifying the starting vertex of the first segment and the ending vertex of each segment, we have enough information to recover the full hike. Note that there are at most

$$k^{O(k/g)},$$

ways of encoding a hike. Then, the number of $2k$-hikes in G_0 is at most

$$n \cdot d \cdot (d-1)^k \cdot 2^{2k} \cdot k^{O(k/g)}.$$

Now we can take $k \approx n^\delta$ for a sufficiently small $\delta = \delta(d) > 0$ and obtain, after taking the $2k$-th root of the above quantity,

$$\rho(B_s) \ \leqslant \ (1+\epsilon) \cdot 2 \cdot \sqrt{(d-1)},$$

when $k = k(n, d, \epsilon)$ is sufficiently large and $c = c(\epsilon)$ is sufficiently small. The extra factor 2 prevent us from obtaining near-Ramanujan bounds with this counting. Nonetheless, the simple counting already allows us to obtain expansion $O(\sqrt{d})$ for lifts sizes as large as $2^{n^{\delta(d)}}$. Moreover, by weakening the expansion guarantee we can obtain lift sizes as large as $2^{n^{\Theta(1)}}$ from this counting and obtain part of Theorem 7.1.2. If we insist on getting a near-Ramanujan bound, we need to compress the traversal history further since storing a string $\{ R, B \}^{2m}$ is too costly and leads to this factor of 2. Note that this string has an equal number of R and B symbols, so it cannot be naively compressed.

To obtain a near-Ramanujan graph, we will take advantage of the special structure of the hike graph (when the walk length is large but not too large) in which most of its vertices have degree exactly 2. These degree 2 vertices are particularly simple to handle in a DFS traversal. For them, we only need to store the next neighbor out of $d-1$ possibilities in G_0 (except possibly for the

first step). In a sequence of backtrackings, if the top of the DFS stack is a degree 2 vertex we know that we are done processing it since no further recursive call will be initiated from it. Then, we simply pop it from the stack. It is for the "rare" at most $\delta \cdot n'$ vertices v of degree $\geqslant 3$ that we need to store how many extra recursive calls t_v we issue from v and a tuple of additional next neighbors $(d_1, \ldots, d_{t_v})$. The total number of such encodings is at most

$$n \cdot d \cdot (d-1)^k \cdot \binom{k+1}{\delta(k+1)} \cdot (d-1)^{\delta(k+1)},$$

which combined with the same previous way of encoding a hike given its graph results in a total number of hike encodings of G_0 of at most

$$n \cdot d \cdot (d-1)^k \cdot \binom{k+1}{\delta(k+1)} \cdot (d-1)^{\delta(k+1)} \cdot k^{O(k/g)},$$

By choosing $\delta = \delta(d, \epsilon)$ sufficiently small and taking $k = k(n, d, \epsilon) \leqslant 2^{\delta \cdot g} \approx n^{O_d(\delta)}$ sufficiently large, we obtain after taking the $2k$-th root

$$\rho(B_s) \ \leqslant \ \sqrt{(d-1)} + \epsilon,$$

indeed leading to a near-Ramanujan bound for lifts as large as 2^{n^δ} in Theorem 7.1.2.

Now we briefly explain how to handle the union bound to ensure that $\rho(B_s(\chi))$ is *simultaneously* small for all $(\ell - 1)$ non-trivial characters (in the decomposition of Fact 7.2.3). This union bound is *standard* when using the trace power method, what is relevant is the trade-off between lift size and walk length. To obtain a high probability guarantee from a guarantee on expectation, it is standard to consider larger walk lengths from which concentration follows from a simple Markov inequality. More precisely, if for some function f , $\mathbf{E}[\rho(B_s(\chi_j))^{2k}] \leqslant f(n, d, g, k)$, then by Markov's inequality,

$$\mathbf{Pr}_{s \in \mathbb{Z}_\ell^{E_0}} \left[\rho(B_s(\chi)) \geqslant 2^{\log_2(\ell)/(2k)} \cdot f(n, d, g, k)^{1/(2k)} \right] \ \leqslant \ \frac{1}{\ell}.$$

Therefore, for $k \geqslant \log_2(\ell)$ sufficiently large, we can union bound over all characters χ and obtain similar bounds as before. As alluded above, this lower bound on the length of the walk depending on the lift size is the reason why we are led to consider much longer walks. To conclude this proof sketch, we need to replace a random signing by a pseudorandom random one. We use ϵ-biased distributions but suitably generalized to abelian groups, e.g., the one by Jalan and Moshkovitz in [JM21]. For our application, it suffices to have the support size of the ϵ-biased distribution polynomial in $1/\epsilon$. We may be taking very large walks on the base graph G_0, so the error of the generator needs to be smaller than $n \cdot d^{2k}$, where k can be as large as $n^{\Theta(1)}$. We note that as long as the degree d is a constant this quantity is at most a polynomial in the size of the *final* lifted graph G since walks of length $O(\log(|V(G)|))$ suffice for any lift size up to full extent of $2^{O(n)}$, for which abelian lifts can be expanding.

7.4 A new encoding for special walks

In this section we will count the total number of singleton-free hikes of a given length on a fixed graph, G. We split the count into two parts. First, we count the number of possible hike

graphs and then, for a given hike graph $\mathcal{H}$, we count the number of hikes that can i.e., yield $\mathcal{H}$ on traversal. Each of these counts is via an encoding argument and therefore we have two kinds of encoding. One for graphs and the other for hikes. In the first part of the section we give two ways of encoding graphs, and in the other half, we encode hikes. Since the first section is a general encoding for subgraphs, we relegate formal definitions related to hikes to a later section.

7.4.1 Graph encoding

Let $\mathcal{H}$ be a subgraph of a fixed d-regular graph G. We wish to encode $\mathcal{H}$ in a succinct way such that given the encoding and G, we can recover $\mathcal{H}$ uniquely. We will give two ways of encoding $\mathcal{H}$. The first one will be generic that works for any subgraph of a d-regular graph. The second encoding takes advantage of the special sparse structure (not too many vertices of degree greater than two). We assume that we have an order on the neighbors of every vertex, and thus, given (v, j), we can access the j^{th} neighbor of v efficiently.

We will do this by encoding a DFS based-traversal of it from a given start vertex. Here, we really need our DFS traversal to be optimal in the sense that the number of times each edge is traversed is at most two and not any higher.

To reconstruct the graph, we reconstruct the traversal and so we need access to two types of data before every step - (1) Is this step recursive or backtracking (2) If it is a recursive step, then which neighbor do we recurse to.

To determine the neighbor of the current vertex we need to move to in a recursive call we need to specify one out $d-1$ possibilities (except in the first step which has d possibilities). This can be specified by a tuple of $(d_1, \ldots, d_{|E(\mathcal{H})|}) \in [d] \times [d-1]^{|E(\mathcal{H})|-1}$ indicating the neighbor. For a backtracking step, we just pop the stack and thus don't need any additional data.

We use two ways to figure out whether a step is recursive or backtracking. The direct way (**Encoding I**) is to just record the sequence in a binary string of length $2|E\mathcal{H}|$. To define the second way, first let a neighbor u of v by called *recursive* if the edge (v, u) is visited by a recursive call from v. A simple observation about backtracking sequences is that it starts when we encounter a vertex that has already been visited or we reach a degree one vertex and ends when we see a visited vertex that has unvisited recursive neighbors. Therefore, (**Encoding II**) we store a string $\sigma \in [d] \times [d-1]^{|V(\mathcal{H})|-1}$ in which σ_i denotes the number of recursive neighbors of the i^{th} visited vertex. To summarize,

GraphEnc$(\mathcal{H})$:

(a) Starting vertex $v_1 \in V(G)$

(b) A sequence of degrees $(d_1, \ldots, d_{|E(\mathcal{H})|}) \in [d] \times [d-1]^{|E(\mathcal{H})|-1}$

(c) Either $\sigma \in \{\, \mathrm{R}, \mathrm{B} \,\}^{2|E(\mathcal{H})|}$ (**Encoding I**) or $\sigma \in [d] \times [d-1]^{|V(\mathcal{H})|-1}$ (**Encoding II**)

Algorithm 7.4.1 (Unpacking Algorithm for **GraphEnc**).
Input **GraphEnc**$(\mathcal{H})$
Output $\mathcal{H}$

- Initialize DFS stack S with v_1
- Initialize $\mathcal{H} = (\{ v_1 \}, \emptyset)$
- Initialize $n, r, t = 1$ // count visited vertices, recursive steps and total steps
- Initialize $ord(v_1) = 1$
- While $S \neq \emptyset$:
  - Let v be the top vertex on the stack S
  - step $=$ StepType(v, t)
  - If step $=$ R (recursive):
    - Assign v_{next} to be d_r^{th} neighbor of v and increment r
    - Add edge $\{ v, v_{\text{next}} \}$ to $\mathcal{H}$
    - If v_{next} is unvisited :
      - Add vertex v_{next} to $\mathcal{H}$
      - $n \leftarrow n + 1$
      - $\text{ord}(v_{\text{next}}) \leftarrow n$
      - push(v_{next}, S)
    - Else if v_{next} is visited, increment t // Next step is backtracking
  - If step $=$ B (backtracking):
    - pop(S)
    - $t \leftarrow t + 1$
- return $\mathcal{H}$

Algorithm 7.4.2 (StepType).
Input (v, t)
Output (Type)
Note - The subroutine to detect the type of step depends on the encoding string σ.

- If σ is from **Encoding I**, return σ_t
- Else, let $j = \text{ord}(v)$
  - If $\sigma_j > 0$ //Check if there are any remaining recursive neighbours
    - Decrement $\sigma_j \leftarrow \sigma_j - 1$
    - return R
  - Else, return B

Counting the encodings

For the first kind of encoding of type, we have 2^{2k} strings of length $2k$ over $\{R, B\}$. The second encoding might seem wasteful in general but it is much better when the graph has special structure that our hike graph will satisfy. We first note that for any vertex v, the number of recursive neighbours $\sigma_v \leqslant \deg_{\mathcal{H}}(v) - 1$ (or $\leqslant \deg_{\mathcal{H}}(v)$ if $v = v_0$).

Definition 7.4.3 (Excess Set)**.** We define a vertex to be an *excess vertex* in $\mathcal{H}$ if $deg_{\mathcal{H}}(v) > 2$ and we define the *excess set* to be the set consisting of such vertices i.e

$$\text{excSet}(\mathcal{H}) \coloneqq |\{\, v \in V(\mathcal{H}) \mid \deg(v) > 2 \,\}|.$$

Lemma 7.4.4. *Let G be a fixed d-regular graph on n vertices. The total number of connected subgraphs $\mathcal{H}$ of G having at most $\leqslant k$ edges is at most*

$$2n \cdot d \cdot (d-1)^{k-1} \cdot 2^{2k}.$$

Moreover, if $\mathcal{H}$ is constrained to have at most two vertices of degree one[5] *and $exc(\mathcal{H}) \leqslant \delta k$, the count is at most*

$$2nk^3 \cdot d \cdot (d-1)^{k-1} \cdot 2^{H_2\left(\frac{\delta}{1-\delta}\right)k} \cdot d^{\delta k}.$$

Proof. We first fix the number of edges as m and we will then sum up the expression for $m \leqslant k$. Algorithm 7.4.1 unambiguously recovers the graph and therefore the number of possible graphs can be counted by counting the number of possible inputs. The number of degree sequences and start vertices are $n \cdot d(d-1)^{m-1}$. The number of σ-strings of encoding I are 2^{2m}. Therefore for a given m, we have $nd \cdot (d-1)^{m-1} \cdot 2^{2m}$ and summing this gives the first claim.

In the second case, the key idea is that for every vertex (except the start) of degree 2, σ_v must be 1. Since $|\text{excSet}(\mathcal{H})| \leqslant \delta m$, almost all of the string σ is filled by 1.

We first pick the number of vertices, say t. There are at most m choices for this. Then, we let the number of excess vertices be j. Summing over all possible j, the number of σ-strings of length t is $\leqslant t^2 \sum_{j=0}^{\delta m} \binom{t}{j} d^j \leqslant t^2 d^{\delta m} \sum_{j=0}^{\delta m} \binom{t}{j} \leqslant t^2 d^{\delta m} 2^{H_2\left(\frac{\delta}{1-\delta}\right)t}$.

Here the first term counts the ways or having or up to two vertices of degree 1, the second counts the ways to choose the excess vertices and the third counts the number of their recursive neighbours. In the last inequality we used that $t = m - exc(\mathcal{H}) \geqslant (1-\delta)m$.

The complete expression for the number of graphs would then be

$$\sum_{m \leqslant k} \left(nd(d-1)^{m-1} \sum_{t=(1-\delta)m}^{m} t^2 d^{\delta m} 2^{H_2\left(\frac{\delta}{1-\delta}\right)t} \right) \;\leqslant\; 2nk^3 \cdot d \cdot (d-1)^{k-1} \cdot 2^{H_2\left(\frac{\delta}{1-\delta}\right)k} \cdot d^{\delta k}.$$

□

[5] We will see later that hike graphs satisfy this strange property

7.4.2 Bounding special walks

A singleton-free k-hike on G defines a subgraph $\mathcal{H}$ such that there at most two vertices of degree 1 (the start vertex and the middle vertex) and the number of edges is at most k as every edge is traversed at least twice. The goal now is to count the possible number of singleton-free k-hikes that yield a fixed subgraph $\mathcal{H}$. Having access to $\mathcal{H}$, we will need to encode the hike in a way similar to the encoding of stale stretches in Chapter 4.

HikeEnc:

(a) $(v_1, \ldots, v_s) \in V(\mathcal{H})^s$,where $s = \lceil 2k/r \rceil$ and r is the bicycle free radius of $\mathcal{H}$.

(b) $(c_1, \ldots, c_s) \in \{0, \pm 1, \cdots, \pm\lfloor r/2 \rfloor\}^s$. Here, c_i denotes the number of times the unique cycle (in the neighborhood of v_i) is to be traversed and the sign indicates the orientation. Since each stretch is of length r and each cycle of length at least 2 we can traverse a cycle at most $\lfloor r/2 \rfloor$ times.

Claim 7.4.5. For any graph $\mathcal{H}$ that is bicycle free at radius r, the number of simple singleton-free k-hikes that have $\mathcal{H}$ as their hike graph is at most $(|rV(\mathcal{H})|)^{\lceil 2k/r \rceil}$.

Proof. Follows from the possible values the encoding **HikeEnc** can take. □

Using Corollary 4.2.7 we can conclude the following corollary:

Corollary 7.4.6. *Let G be a d regular graph on n vertices bicycle free at radius r. Let $\mathcal{H}$ be a subgraph with at most two vertices of degree one on n_0 vertices where $n_0 = e^{\delta r - 1}$ for some $\delta \leqslant 1/10$. Then,*

$$\text{excSet}(\mathcal{H}) \leqslant 2\delta n_0 + 2.$$

We can now state the main bound on the number of special walks.

Lemma 7.4.7. *Let G be a d regular graph, with $d \geqslant 3$, on n vertices bicycle free at radius r. Then, the total number of singleton free $(k-1)$-hikes on G is at most*

$$\left(2^{\gamma_1}\sqrt{d-1}\right)^{2k} \text{ where } \gamma_1 = 1 + \frac{\log(nrk)}{2k} + \frac{\log(rk)}{r}.$$

If we assume that $3 \leqslant k \leqslant e^{\delta r}$, then it is at most

$$\left(2^{\gamma_2}\sqrt{d-1}\right)^{2k} \text{ where } \gamma_2 = \frac{\log(16nk^3rd)}{2k} + \frac{\log(rk)}{r} + H_2(5\delta)/2 + \delta \log d.$$

Proof. Any singleton-free $(k-1)$-hike defines a connected graph $\mathcal{H}$ with at most $k-1$ edges and therefore at most $k-1$ vertices. If there is no backtracking step then all vertices except the start have degree at least two. Else, the end point of one of the backtracking step may have degree 1. Thus there are at most 2 vertices of degree one. When k is unbounded, we use the bound from the first encoding i.e. Lemma 7.4.4 and combine it with the number of possible hikes on this from Claim 7.4.5 to get

$$\begin{aligned}
&\leqslant 2n \cdot d \cdot (d-1)^{k-2} \cdot 2^{2(k-1)} (r(k-1))^{\frac{2(k-1)}{r}+1} \\
&\leqslant (nrk) \cdot (d-1)^k \cdot 2^{2k} (rk)^{\frac{2k}{r}}
\end{aligned}$$

$$\leqslant \left(2 \cdot 2^{\log(nrk)/2k} 2^{\frac{\log(rk)}{r}}\right)^{2k} (d-1)^k$$
$$\leqslant \left(2^{\gamma_1}\sqrt{d-1}\right)^{2k}.$$

The assumption on k lets us use Corollary 7.4.6 which when combined with Lemma 7.4.4 gives us the bound on the number of such graphs as $4nk^2d\cdot(d-1)^{k-1}\cdot\binom{k}{2\delta k+1}\cdot d^{2\delta k+1}$. Combining with the number of possible hikes on this from Claim 7.4.5, we get the total number of singleton-free k-hikes bounded by

$$\leqslant 4n(k-1)^2 \cdot d \cdot (d-1)^{k-2} \cdot \binom{k-1}{2\delta(k-1)+2} \cdot d^{2\delta(k-1)+2}(r(k-1))^{\frac{2k-2}{r}+1}$$
$$\leqslant (16nk^3rd)(d-1)^k \cdot 2^{H_2(5\delta)k} \cdot d^{2\delta k}(rk)^{\frac{2k}{r}}$$
$$\leqslant \left(2^{\log(16nk^3rd))/2k} d^{\delta} 2^{\frac{\log(rk)}{r}} 2^{H_2(5\delta)/2}\right)^{2k} (d-1)^k$$
$$\leqslant \left(2^{\gamma_2}\sqrt{d-1}\right)^{2k}.$$

□

7.5 Explicit expanding abelian lifts

In this section, we will use the bound on singleton-free hikes obtained in the last section to bound the eigenvalue of the lifted graph. We first handle non-singleton free hikes and show that they can be easily bounded by the ε-biased property of the distribution of the signings. We then formalize the construction by instantiating it using an expander from MOP having large bicycle-free radius and then bring the bounds together.

7.5.1 Generalizing the trace power nethod

We now show that the the problem of bounding the spectral radius of the signed non-backtracking operator reduces to counting singleton free hikes. This reduction is a straightforward generalization of the argument Proposition 4.3.3 for $\mathbb{Z}_2$ to any abelian group.

Let $B_s(\chi)$ (as defined in Definition 7.2.1) be the signed non-backtracking operator with respect to a signing and a non-trivial character χ and $\rho(B_s)$ denote its spectral radius. The goal is to bound the largest eigenvalue of $B_s(\chi)$.

The signing s is drawn from some distribution $\mathcal{D}$ and we wish to show via the probabilistic method that there exists a signing in $\mathcal{D}$ for which $\rho(B_s(\chi))$ is small for any set of $(l-1)$ non-trivial characters χ. We will use a first-order Markov argument and therefore wish to bound $\mathbf{E}_{s\sim\mathcal{D}}[\mathrm{tr}(B_s^k(B_s^*)^k)]$. Writing it out we get,

$$T_\chi(s) = \mathrm{tr}((B_s^*)^k B_s^k) = \sum_{e\in\vec{E}} \left((B_s^*)^k B_s^k e\right)_e$$
$$= \sum_{(e_0,\cdots,e_{2k})} B(e_0,e_1)\cdots B(e_{k-1},e_k)B^*(e_k,e_{k+1})\cdots B^*(e_{2k-1},e_{2k})$$

$$= \sum_{(e_0,\cdots,e_{2k})} \chi(s(e_1))\cdots\chi(s(e_k))\chi^*(s(e_k))\cdots\chi^*(s(e_{2k-1}))$$
$$= \sum_{(e_0,\cdots,e_{2k})} \chi(s(e_1))\cdots\chi(s(e_{k-1}))\chi^*(s(e_{k+1}))\cdots\chi^*(s(e_{2k-1})).$$

Notice that e_0, e_k don't appear in the term and so we define $\mathcal{H}_{k-1}$ as the multiset of all tuples $(e_1, \ldots, e_{k-1}, e_{k+1}, \ldots, e_{2k-1})$ appearing in the support of this summation. We denote each term in the summation above by $\chi_w(s)$ where $w \in \mathcal{H}_{k-1}$. It follows directly from the definition that each $w \in \mathcal{H}_{k-1}$ defines a $(k-1)$-hike. Also observe that, any tuple appears at most $(d-1)^2$ times as given a tuple w, we have at most $(d-1)$ choices for each e_0, e_k. Let $\mathcal{H}^s_{k-1}$ denote the singleton-free hikes in $\mathcal{H}_{k-1}$. We can split $T_\chi(s) = T_1(s) + T_2(s)$ where

$$T_1(s) = \sum_{w\in\mathcal{H}^s_{k-1}} \chi_w(s), \quad T_2(s) = \sum_{w\notin\mathcal{H}^s_{k-1}} \chi_w(s).$$

We now define ε-biased distributions that will be the key pseudorandomness tool.

Definition 7.5.1 (Bias)**.** Given a distribution $\mathcal{D}$ on a group H and a character χ, we can define the bias of $\mathcal{D}$ with respect to χ as $bias_\chi(\mathcal{D}) := |\mathbf{E}_{h\sim\mathcal{D}}\chi(h)|$ and the bias of $\mathcal{D}$ as $\mathrm{bias}(\mathcal{D}) = \max_\chi bias_\chi(\mathcal{D})$, where the maximization is over non-trivial characters.

Lemma 7.5.2. *Let $\mathcal{D} \subseteq H^{E(G)}$ be an ν-biased distribution and let $w \notin \mathcal{H}^s_{k-1}$ be a singleton-hike i.e. there is an edge that is travelled exactly once. Then, $|\mathbf{E}_{s\sim\mathcal{D}}\chi_w(s)| \leqslant \nu$.*

Proof. Let the set of distinct edges in w be $\{e_1, \cdots, e_r\}$ and let edge e_i be travelled t_i times where t_i takes the sign into account.[6]Let e_j be the edge traversed exactly once. Then, $t_j = \pm 1$. Now, we can rewrite $\chi_w(s) = \prod_{i=1}^r \chi(s(e_i))^{t_i}$ and it can be extended to a character on $H^{E(G)}$. Since $t_j = \pm 1$, this character is non-trivial and the claim follows from the ν-biased property. □

Lemma 7.5.3 (Analog of Corollary 4.3.11)**.** *Let G be a d-regular graph on n-vertices, $\varepsilon < 1$ be a fixed constant, ℓ be a parameter, $H \subseteq Sym(\ell)$ be an abelian group and $\mathcal{D} \subseteq H^m$ be an ν-biased distribution such that $\nu \leqslant (nld^2)^{-1}.\left(\frac{\varepsilon}{d}\right)^{2k}$.*

Assume that the number of singleton-free $(k-1)$-hikes is bounded by $(2^\gamma\sqrt{d-1})^{2k}$. Then for any non-trivial character χ of H^m, we have that except with probability at most $1/\ell$ over $\mathcal{D}$, $\rho(B(\chi)) \leqslant 2^{\gamma'}\sqrt{d-1} + \epsilon$ where $\gamma' = \gamma + \frac{\log(\ell d^2)}{2k}$.

Proof. By the decomposition above, we have $T(s) = T_1(s) + T_2(s)$. As each term in the expression is of the form $\chi(h)$ and as remarked earlier, all the characters are roots of unity so $|\chi(s(e))| = 1$. Thus, $|T_1(s)| \leqslant |\pi^{-1}\left(\mathcal{H}^*_{k-1}\right)| \leqslant (d-1)^2|\mathcal{H}^*_{k-1}|$

$$\begin{aligned}\mu := |\mathbf{E}_{s\sim\mathcal{D}}T| &= |\mathbf{E}T_1 + \mathbf{E}T_2| \\ &\leqslant |\mathbf{E}T_1| + |\mathbf{E}T_2| \\ &\leqslant |\mathcal{H}^s_{k-1}| + \sum_{w\notin\mathcal{H}^s_{k-1}} |\mathbf{E}_{s\sim\mathcal{D}}\chi_w(s)|\end{aligned}$$

[6]Let e_i appear f_1 times in the first $k-1$ steps and b_1 times in the next $(k-1)$ steps. Similarly let e_i^T which is the reverse direction of e appear f_2 times in the first $k-1$ steps and b_2 times in the next $(k-1)$ steps. Then, $t_i = f_1 + b_2 - f_2 - b_1$.

$$\leqslant d^2(2^\gamma\sqrt{d-1})^{2k} + \nu|\mathcal{H}_{k-1}|$$
$$\leqslant d^2(2^\gamma\sqrt{d-1})^{2k} + \nu nd^{2k+2}.$$

Here we have used the observation that $|\mathcal{H}^s_{k-1}| \leqslant (d-1)^2|\{\text{Singleton-free } (k-1)\text{-hikes}\}|$ and Lemma 7.5.2. The bound on $|\mathcal{H}_{k-1}|$ is trivial as we have nd choices for the starting edge and a walk of length of $2k+1$. Since T is a non-negative random variable, we apply Markov to conclude that $T \leqslant \mu\ell$ with probability at most $1/\ell$.

$$\begin{aligned}
\rho(B_s(\chi)) \leqslant T^{1/2k} < (\mu\ell)^{1/2k} &\leqslant \left(d^2\ell\left(2^\gamma\sqrt{d-1}\right)^{2k} + \nu\ell nd^{2k+2}\right)^{1/2k} \\
&\leqslant (d^2\ell)^{1/2k}2^\gamma\sqrt{d-1} + \left(\nu\ell nd^{2k+2}\right)^{1/2k} \\
&\leqslant 2^{\gamma'}\sqrt{d-1} + (\nu\ell nd^2)^{1/2k}d \\
&\leqslant 2^{\gamma'}\sqrt{d-1} + \frac{\varepsilon}{d}d \\
&\leqslant 2^{\gamma'}\sqrt{d-1} + \varepsilon.
\end{aligned}$$

□

7.5.2 Combining all the ingredients

Before we instantiate the explicit construction of abelian lifted expanders leading to Theorem 7.1.2, we will need two tools. The first one is an explicit construction of expander graphs to be used as base graphs in the lifting operation, which follows from Theorem 4.1.1.

The second tool is a ν-biased distribution for abelian groups (having a sample space depending polynomial on $1/\nu$). In particular, we use a recent construction by Jalan and Moshkovitz.

Theorem 7.5.4. *[JM21] Given the generating elements of a finite abelian group H and an integer $m \geqslant 1$ and $\nu > 0$, there is a deterministic polynomial time algorithm that constructs subset $S \subseteq H^m$ with size $O\left(\frac{m\log(H)^{O(1)}}{\nu^{2+o(1)}}\right)$ such that the uniform distribution over S is ν-biased.*

We are now ready to prove our main result.

Proof of Theorem 7.1.2. Construct G_0 on $n \leqslant n' \leqslant 2n$ vertices for given (d,ϵ) using Theorem 4.4.9 which is bycicle-free at radius $r \geqslant c\log_{d-1} n'$.

- **Regime 1 (Sub-Exponential)** - Here shorter walks will suffice and we will use the bound on γ_2 from Lemma 7.4.7. To get Near-Ramanujan, we need $\gamma' = \gamma_2 + \frac{\log(d^2\ell)}{2k} = \gamma_2' + \frac{\log(\ell)}{2k}$ to be vanishing with ε. Observe that when $k = \omega(\log n)$, γ_2 is bounded by $o(1) + \left(2\sqrt{\delta} + \delta\log d\right)$. We pick δ small enough and assume that $n' \geqslant N(\varepsilon, d)$ such that $\gamma_2' \leqslant \frac{2\varepsilon}{\sqrt{d-1}}$. In the bounded k regime we can pick $k < e^{\delta r}$. Since, $\frac{\log(\ell)}{2k}$ must also be vanishing in ε, this forces $\log(\ell) \leqslant \varepsilon k \leqslant \varepsilon e^{\delta r}$. This explains the bound on ℓ.
- **Regime 2 (Moderately-Exponential)** - Here ℓ is larger and so we pick $k = \log\ell$. Now, we need to use γ_1 which we recall is $1 + \frac{\log k}{r} + o(1)$. Thus, $\gamma' = (\gamma_1 + \frac{\log d^2}{k}) + \frac{\log\ell}{k} \leqslant 3/2 + \frac{\log k}{r}$. Since, $r = c\log_{d-1}(n')$, to get non-trivial expansion $k \leqslant n^{c/2}$ which explains the bound on the exponent δ.

The precise parameters are as follows

Regime	δ	k	ν	γ'
1	$O\left(\frac{\varepsilon^2}{d}\right)$	$\frac{10\sqrt{d-1}}{\varepsilon}\max(\log\ell, \log n)$	$(nld^2)^{-1}\left(\frac{\varepsilon}{3d}\right)^{2k} = (n\ell)^{c_{d,\varepsilon}}$	$\frac{2\varepsilon}{3\sqrt{(d-1)}}$
2	$\leqslant c/2$	$\log\ell = n^{\delta}$	$(n\ell d^2)^{-1}\left(\frac{1}{3d}\right)^{2k} = (n\ell)^{c_d}$	$2 + \frac{\delta}{c}\log(d-1)$

Construct a ν-biased distribution $\mathcal{D}$ using theorem 7.5.4. These two constructions take $\mathrm{poly}(n, \ell)$ time.

From corollary 7.2.5, we have to analyze $B(\chi)$ for $\ell - 1$ non-trivial characters χ that appear in this decomposition. The largest eigenvalue is clearly given by $B(1)$ which is $d - 1$. For the second largest, $\lambda_2(B(1)) \leqslant \sqrt{d-1} + \varepsilon$ by the property of the base graph G. Since we have the bicycle-free property, we can use corollary 7.5.3 to conclude that for any non-trivial characters we have except with probability at most $1/\ell$

- **Regime 1** - $\rho(B(\chi)) \leqslant 2^{\gamma'}\sqrt{d-1} + \epsilon/3 \leqslant \sqrt{d-1} + \varepsilon$.
- **Regime 2** - $\rho(B(\chi)) \leqslant 2^{\gamma'}\sqrt{d-1} + 1 \leqslant 2 \cdot 2^2 d^{\delta/c}\sqrt{d-1} \leqslant \varepsilon d$ when $d \geqslant \left(\frac{8}{\varepsilon}\right)^{\frac{2c}{c-2\delta}}$.

Using the fact Fact 7.2.3, we assume that the decomposition has exactly one trivial character (say, χ_1) and $(l-1)$ non-trivial characters. Then for the trivial character $\rho(B_{G_0(s)}) = \rho(B(\chi_1)) = d - 1$ and thus, $\rho_2(B) = \max\left\{\lambda(G_0), \max_{i=2}^{\ell}\rho(B(\chi_i))\right\}$.

Since the bound holds for any non-trivial χ except with probability $1/\ell$ we take a union bound over these $\ell - 1$ characters we get that there is a labelling $s \in D$ such that the bound holds for $\rho(B(\chi_i))$ and thus for $\lambda(B_{G_0(s)})$. By Fact 7.2.2, we get that $\lambda(G) \leqslant 2\rho_2(B_G)$ which satisfies the bounds we need.

We can brute force through each $s \in \mathrm{supp}(\mathcal{D})$ to find an s such that the lifted graph $G = G_0(s)$ has the required spectral gap. Checking this is a simple linear algebraic task and can be done in time cubic in $n\ell$. Therefore, the total time taken is $\mathrm{poly}(n, \ell)$. □

7.6 Explicit quantum and classical codes

We now briefly recall the construction of quantum LDPC codes as in [PK21] and show how our results derandomize it. The construction is as follows. Let G be a d-regular graph (on $n\ell$ vertices) such that G is a $(\mathbb{Z}_\ell, \ell)$-lift of a graph on n-vertices. Let $\mathcal{C}_0 \subseteq \mathbb{F}_2^d$ be a binary linear code (of block length d). Let B denote the bipartite graph of the Tanner code $\mathrm{T}(G, \mathcal{C}_0)$ and let F denote the cycle graph on ℓ vertices. They define the lifted product $\mathrm{LP}(B, F)$ of B and F which is a variation of the usual tensor product and is also equivalent to the twisted product in [HHO21]. The main result of [PK21] is the following.

Theorem 7.6.1 ([PK21])**.** *Let G be $(\mathbb{Z}_\ell, \ell)$-lift of a d regular graph on n-vertices with $\lambda_2(G) \leqslant \varepsilon \cdot d$. Let $\mathcal{C}_0 \subseteq \mathbb{F}_2^d$ and its dual attain the Gilbert–Varshamov bound. If $\varepsilon > 0$ is sufficiently small and d is a sufficiently large constant, then*

- $\mathrm{T}(G, \mathcal{C}_0)$ *is a good quasi-cyclic LDPC code of blocklength $\Theta(n\ell)$ and circulant size $\Theta(\ell)$.*
- *The quantum lifted product code $\mathrm{LP}(B, F)$ is LDPC and has distance $\Theta_{\epsilon,d}(\ell)$ and dimension $\Theta(n)$.*

To achieve these, [PK21] picks a d-regular expander on n vertices and creates a random ℓ-lift i.e. where each signing is chosen uniformly at random from $\mathbb{Z}_\ell$. The final graph is expanding with high probability from the results of [ACKM19] (Theorem 7.1.3). The distance achieves the almost-linear bound only when the lift is large and thus lifts of exponential size are preferred. By the upper bound in Theorem 7.1.3, better than exponential size lifts break expansion for abelian groups.

For this application, the constant degree regime is important for two reasons. The locality of the code is essentially d and thus it has to be constant for it to be LDPC. Moreover, the code $\mathcal{C}_0 \subseteq \mathbb{F}_2^d$ can be easily constructed via brute-force search since d is constant.

While the corollary follows in a straightforward manner from our main results, we show the computations for completeness.

Corollary 7.6.2 ([PK21], Theorem 7.1.2)**.** *We have explicit polynomial time construction of each of the following,*

1. *Good quasi-cyclic LDPC code of block length N and any circulant size up to $N/\mathrm{polylog}(N)$.*
2. *Quantum LDPC code with distance $\Omega(N/\mathrm{polylog}(N))$ and dimension $\Omega(\mathrm{polylog}(N))$.*
3. *Quantum LDPC code with distance $\Omega(N^{1-\alpha})$ and dimension $\Theta(N^\alpha)$ for every constant $\alpha > 0$.*

Proof. We always work in the constant degree regime so $\mathcal{C}_0 \subseteq \mathbb{F}_2^d$ can be found by brute-force. Let $\ell \leqslant 2^{n^{\delta_0}}$ with some fixed $\delta_0 \in (0,1)$. We can explicitly construct G which is a $(\mathbb{Z}_\ell, \ell)$-lift by Theorem 7.1.2 and by [PK21], $\mathrm{T}(G, \mathcal{C}_0)$ has circulant size $\Theta(\ell)$ and $\log(N) \leqslant \log n + n^{\delta_0} \leqslant 2n^{\delta_0}$ (for n sufficiently large) and thus, $\ell = O\left(N/(\log N)^{1/\delta_0}\right)$. Therefore, the construction works for any size less than $N/(\log N)^{1/\delta_0}$. This calculation also shows that we get quantum LDPC codes for any distance less than $N/(\log N)^{1/\delta_0}$. So for a given α, we take a base graph on $n = N^\alpha$ vertices and construct a $\ell = N^{1-\alpha} = n^{1/\alpha - 1}$ lift. For any α, this is a polynomial sized-lift and can be done via Theorem 7.1.2. □

Chapter 8

Open Problems and Closing Remarks

We conclude this thesis by discussing some open problems that fit the template of the three fundamental questions of this thesis.

Open Question 1. *Can we build on these results to obtain near-Ramanujan graphs with other combinatorial properties (e.g. high girth)?*

For a lot of applications of expander graphs it is important that the expander also has extra combinatorial properties. As we saw in Chapter 6, an example of such a property is high girth, which has implications in the decodability of some families of error correcting codes. This motivates the next question:

Open Question 2. *Can we improve Theorem 6.1.6 to obtain high ($\Theta(\log_{d-1} n)$) girth?*

Something like this could be proved by showing that when 2-lifting a graph with large enough girth, with sufficiently high probability the girth of the resulting graph increases. This would boost the girth of the graph generated by the first step of the construction of [MOP20a] during the repeated 2-lift step. However, it is unclear if this can be done. Alternatively, one could show that bicycle-freeness increases with good probability as we 2-lift, but this is also unclear.

A different strategy would be to find a different way to derandomize Theorem 1.1.16 such that the we can generate a starter graph of larger size. However, it is unclear if this strategy could work since the tool used to derandomize this, namely (δ, k)-wise uniform permutations, cannot be improved to derandomize this to the required extent.

Open Question 3. *Can we obtain a combinatorial strongly explicit construction of near-Ramanujan graphs?*

The zig-zag product of Reingold-Vadhan-Wigderson [RVW02] is the only known *combinatorial* construction of strongly explicit expander graphs. It would be interesting to find other strongly explicit constructions with better expansion, perhaps based on the results of this section.

Open Question 4. *Are there explicit Ramanujan graphs of all degrees d?*

This is the "holy grail" of the field of expanders, the question that has been asked since the '70s. Even the existence of such graphs for many degrees (non prime powers) is open. The results in Chapter 4 got us closer to this answer, but a completely different approach will be needed to fully answer it.

Open Question 5. *What can we say about the expansion of non-abelian lifts?*

One of the reasons why shift lifts were studied in the context of expanders was as a potential

way to build Ramanujan graphs of all degrees [CV17]. Suppose we could prove the following: There exists a shift k-lift that maintains the Ramanujan property of d-regular graphs on n vertices for all n. Then, we could start with a complete d-regular graph K_{d+1} and then there would be a shift k-lift for $k \sim n$ that lifts this starting graph to a Ramanujan graph with n vertices. Since there are at most k^{d^2} shift lifts, a simple brute force algorithm would work in polynomial time.

Unfortunately, [ACKM19] showed that this is impossible since for $k = 2^{\Omega(n_0 d)}$ (where n_0 is the number of vertices of the starting graph) there is no expanding shift lift. Even more, they show the same also applies for any abelian lift. This closes a lot of possibilities, however, we know that for general graph lifts, a generalization [HPS18] of [MSS15a] shows that there is always a one-sided Ramanujan k-lift of any graph. This result is only existential and it is hard to make explicit, but it suggests the question: what can we say about the expansion of other non-abelian group lifts? Can we find a good expanding non-abelian group that is easy to derandomize, just like shift lifts?

Recently, Panteleev and Kalachev [PK22] used non-abelian group lifts in their breakthrough result on asymptotically good quantum and locally testable classical LDPC codes.

Open Question 6. *Can we obtain explicit constructions of graphs with other notions of expansion (e.g. lossless expanders)?*

There are other notions of expansion we haven't explored in this thesis. A well known example is *vertex expansion*, which we denote by $\Psi_V(G, k)$ and define as follows:

$$\Psi_V(G, k) = \min_{\substack{S \subset V \\ |S| \leqslant k}} \frac{|N(S)|}{|S|}.$$

One can show that for every $\delta > 0$ there exists $\varepsilon > 0$ such that for almost every n-vertex d-regular graph G we have $\Psi_V(G, \varepsilon n) \geqslant d - 1 - \delta$. This is virtually optimal, since for a connected subset S of vertices we have $\Psi_V(G, |S|) \leqslant d - 1 + 2/|S|$. Graphs that achieve this optimal bound are known as *lossless expanders*. These types of graphs have many applications in areas like distributed routing algorithms, expander-based linear codes and even lower bound complexity (see [HLW06] for a list of references).

So we know that random graphs are lossless expanders with high probability, however the best known explicit result is by Kahale [Kah95], who showed that n-vertex Ramanujan graphs G satisfy $\Psi_V(G, \varepsilon n) \geqslant (d/2) \cdot (1 - o_d(1)) \cdot (1 - O(\log d / \log(1/\varepsilon)))$. Kahale also showed that this is tight, i.e. there are Ramanujan graphs with vertex expansion at most $d/2$. Recently, McKenzie and Mohanty [MM21] showed that that high-girth Ramanujan graphs are also not necessarily lossless expanders. Also recently, Kamber and Kaufman [KK22] disproved the conjecture that certain number theoretical constructions of Ramanujan graphs (namely the result from Morgenstern [Mor94]) are lossless expanders.

Capalbo-Reingold-Vadhan-Wigderson [CRVW02] took a major step in answering this question by providing an explicit construction of a bipartite graph whose left bipartition losslessly expands onto the right one for sets of linear size. Their result is based on the zig-zag product of Reingold-Vadhan-Wigderson [RVW02]. Fortunately, some of the aforementioned applications of lossless expanders only need this notion of bipartite lossless expansion. Another result of interest is the construction by Alon-Capalbo [AC02] of *explicit unique-neighbor expanders* for some degrees, a related notion of vertex expanders (optimal unique-neighbor expanders are

lossless expanders).

But the main question remains, can we build explicit lossless vertex expanders (for any parameters n or d)? A simpler question is: can we merely improve on Kahale's result of vertex expansion of about $d/2$?

Bibliography

[AA15] Scott Aaronson and Andris Ambainis. Forrelation: a problem that optimally separates quantum from classical computing. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*, pages 307–316, 2015. 5.2.2

[ABW10] Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing*, pages 171–180, 2010. 5.1

[AC88] N. Alon and F. R. K. Chung. Explicit construction of linear sized tolerant networks. In *Proceedings of the First Japan Conference on Graph Theory and Applications (Hakone, 1986)*, volume 72, pages 15–19, 1988. doi:10.1016/0012-365X(88)90189-6. 1.1

[AC02] Noga Alon and Michael Capalbo. Explicit unique-neighbor expanders. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 73–79. IEEE, 2002. 8

[ACKM19] Naman Agarwal, Karthekeyan Chandrasekaran, Alexandra Kolla, and Vivek Madan. On the expansion of group-based lifts. *SIAM J. Discrete Math.*, 33(3):1338–1373, 2019. doi:10.1137/17M1141047. 7.1, 7.1.3, 7.1.2, 7.6, 8

[ADR82] Alain Aspect, Jean Dalibard, and Gérard Roger. Experimental test of Bell's inequalities using time-varying analyzers. *Physical Review Letters*, 49(25):1804–1807, 1982. 5.2.7

[AFH15] Omer Angel, Joel Friedman, and Shlomo Hoory. The non-backtracking spectrum of the universal cover of a graph. *Transactions of the American Mathematical Society*, 367(6):4287–4318, 2015. 3.2, 3.2, 5.2.21, 5.5.1, 5.5.1

[AGHP92] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures & Algorithms*, 3(3):289–304, 1992. 3.4

[AGS21] Noga Alon, Shirshendu Ganguly, and Nikhil Srivastava. High-girth near-Ramanujan graphs with localized eigenvectors. *Israel J. Math.*, 246(1):1–20, 2021. doi:10.1007/s11856-021-2217-y. 6.1, 6.1.1, 6.2, 6.2.2, 6.2.9

[AHL02] Noga Alon, Shlomo Hoory, and Nathan Linial. The Moore bound for irregular graphs. *Graphs and Combinatorics*, 18(1):53–57, 2002. 4.2.6, 5.6.3, 7.3

[AL13] Noga Alon and Shachar Lovett. Almost k-wise vs. k-wise independent permuta-

tions, and uniformity for general group actions. *Theory of Computing*, 9:559–577, 2013. 3.4, 3.4.5, 3.4.6

[Alo86] Noga Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986. 1.1, 1.1.2

[Alo21] Noga Alon. Explicit expanders of every degree and size. *Combinatorica*, 41(4):447–463, 2021. doi:10.1007/s00493-020-4429-x. 1.1.1, 1.1.14, 1.1, 6.1.1, 6.2.2

[AM85] N. Alon and V. D. Milman. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *J. Combin. Theory Ser. B*, 38(1):73–88, 1985. doi:10.1016/0095-8956(85)90092-9. 1.1

[Amb06] Andris Ambainis. Polynomial degree vs. quantum query complexity. *Journal of Computer and System Sciences*, 72(2):220–238, 2006. 5.2.2

[APV16] Andris Ambainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. Sensitivity versus certificate complexity of Boolean functions. In *Proceedings of the 11th Annual Computer Science Symposium in Russia*, pages 16–28, 2016. 5.2.2

[AR94] Noga Alon and Yuval Roichman. Random cayley graphs and expanders. *Random Struct. Algorithms*, 5(2):271–285, 1994. doi:10.1002/rsa.3240050203. 7.1

[Bab74] László Babai. On the minimum order of graphs with given group. *Canadian Mathematical Bulletin*, 17(4):467–470, 1974. doi:10.4153/CMB-1974-082-9. 7.1

[Bas92] Hyman Bass. The Ihara–Selberg zeta function of a tree lattice. *International Journal of Mathematics*, 3(6):717–797, 1992. 3.2

[BC78] Edward Bender and Rodney Canfield. The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory. Series A*, 24(3):296–307, 1978. 3.3, 6.3.1

[BDH18] Gerandy Brito, Ioana Dumitriu, and Kameron Decker Harris. Spectral gap in random bipartite biregular graphs and its applications. *arXiv preprint arXiv:1804.07808*, 2018. 5.6

[BE21] Nikolas P. Breuckmann and Jens N. Eberhardt. Balanced Product Quantum Codes. *IEEE Transactions on Information Theory*, 67(10):6653–6674, 2021. arXiv:2012.09271, doi:10.1109/TIT.2021.3097347. 7.1, 7.1.2, 2

[Bel64] John Bell. On the Einstein Podolsky Rosen paradox. *Physics Physique Fizika*, 1(3):195–200, 1964. 5.2.7

[BFKL93] Avrim Blum, Merrick Furst, Michael Kearns, and Richard Lipton. Cryptographic primitives based on hard learning problems. In *Proceedings of the 13th Annual International Cryptography Conference*, pages 278–291, 1993. 5.1

[BKM17] Jess Banks, Robert Kleinberg, and Cristopher Moore. The Lovász Theta function for random regular graphs and community detection in the hard regime. In *Proceedings of the 21st Annual International Workshop on Randomized Techniques in*

Computation, volume 81, pages 28:1–28:22, 2017. 5.1

[BL06] Yonatan Bilu and Nathan Linial. Lifts, discrepancy and nearly optimal spectral gap. *Combinatorica*, 26(5):495–519, 2006. 1.1.1, 1.1.12, 1.1, 3.3, 4.1.1, 4.1.2, 5.2.16, 7.1.1

[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. 5.1.2

[BM06] L.M.J. Bazzi and S.K. Mitter. Some randomized code constructions from group actions. *IEEE Transactions on Information Theory*, 52(7):3210–3219, 2006. doi:10.1109/TIT.2006.876244. 7.1

[BMMN13] Mark Braverman, Konstantin Makarychev, Yury Makarychev, and Assaf Naor. The Grothendieck constant is strictly smaller than Krivine's bound. *Forum of Mathematics. Pi*, 1:e4, 42, 2013. 5.2.1

[Bol80] Béla Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1(4):311–316, 1980. 3.3, 6.1

[Bol01] Béla Bollobás. *Random Graphs*. Cambridge University Press, second edition edition, 2001. 3.3

[Bor15] Charles Bordenave. A new proof of Friedman's second eigenvalue theorem and its extension to random lifts. *arXiv preprint arXiv:1502.04482*, 2015. 5.6, 5.6.1

[Bor19] Charles Bordenave. A new proof of Friedman's second eigenvalue theorem and its extension to random lifts. Technical Report 1502.04482v4, arXiv, 2019. To appear in Annales scientifiques de l'École normale supérieure. 1.1.2, 1.1.16, 3.2, 3.3, 4.1.1, 4.3, 4.3, 4.4, 4.4, 4.4.1, 4.4.2, 4.4.5, 4.4.2, 4.4.2, 5.1.2, 6.3, 6.3.1

[BSS05] László Babai, Amir Shpilka, and Daniel Stefankovic. Locally testable cyclic codes. *IEEE Trans. Inf. Theory*, 51(8):2849–2858, 2005. doi:10.1109/TIT.2005.851735. 7.1

[BT11] Avraham Ben-Aroya and Amnon Ta-Shma. A combinatorial construction of almost-Ramanujan graphs using the zig-zag product. *SIAM Journal on Computing*, 40(2):267–290, 2011. 1.1.11, 1.1

[CGHV15] Endre Csóka, Balázs Gerencsér, Viktor Harangi, and Bálint Virág. Invariant Gaussian processes and independent sets on regular graphs of large girth. *Random Structures & Algorithms*, 47(2):284–303, 2015. 5.1

[Chi92] Patrick Chiu. Cubic Ramanujan graphs. *Combinatorica*, 12(3):275–285, 1992. 1.1.1, 1.1

[CHSH69] John Clauser, Michael Horne, Abner Shimony, and Richard Holt. Proposed experiment to test local hidden-variable theories. *Physical Review Letters*, 23(15):880–884, 1969. 5.2.7, 5.2.7

[CHTW04] Richard Cleve, Peter Høyer, Benjamin Toner, and John Watrous. Consequences

and limits of nonlocal strategies. In *Proceedings of the 19th Annual Computational Complexity Conference*, pages 246–249, 2004. 5.2.6

[Cla06] Pete Clark. Ramanujan graphs and Shimura curves. Retrieved from `http://alpha.math.uga.edu/~pete/ramanujanrevisited.pdf`, 2006. 1.1.1, 1.1

[CM08] Sebastian M. Cioabă and M. Ram Murty. Expander graphs and gaps between primes. *Forum Mathematicum*, 20(4):745–756, 2008. 1.1.1, 1.1.13, 1.1.1, 1.1

[Coh16] Michael Cohen. Ramanujan graphs in polynomial time. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, pages 276–281, 2016. 1.1.1, 1.1.15, 1.1

[Con] Keith Conrad. Simultaneous Ccommutativity Of Operators. `https://kconrad.math.uconn.edu/blurbs/linmultialg/simulcomm.pdf`. [Online; accessed June-2022]. 7.2.1

[CPW69] C.L. Chen, W.W. Peterson, and E.J. Weldon. Some results on quasi-cyclic codes. *Information and Control*, 15(5):407–423, November 1969. `doi:10.1016/s0019-9958(69)90497-5`. 7.1

[CRVW02] Michael Capalbo, Omer Reingold, Salil Vadhan, and Avi Wigderson. Randomness conductors and constant-degree lossless expanders. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pages 659–668. ACM, New York, 2002. `doi:10.1145/509907.510003`. 8

[CS96] A. R. Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54:1098–1105, Aug 1996. `doi:10.1103/PhysRevA.54.1098`. 7.1

[CV17] Karthekeyan Chandrasekaran and Ameya Velingker. Shift lifts preserving Ramanujan property. *Linear Algebra Appl.*, 529:199–214, 2017. `doi:10.1016/j.laa.2017.04.031`. 7.1, 8

[CW04] Moses Charikar and Anthony Wirth. Maximizing quadratic programs: extending Grothendieck's Inequality. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 54–60, 2004. 5.2.1

[Dah14] Xavier Dahan. Regular graphs of large girth and arbitrary degree. *Combinatorica*, 34(4):407–426, 2014. `doi:10.1007/s00493-014-2897-6`. 6.1

[dlHM06] Pierre de la Harpe and Antoine Musitelli. Expanding graphs, Ramanujan graphs, and 1-factor perturbations. *Bulletin of the Belgian Mathematical Society — Simon Stevin*, 13(4):673–680, 2006. 1.1.1, 1.1

[DMO+19a] 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 30th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2305–2321, 2019. 4.1.1

[DMO+19b] 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 30th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 2305–2321, 2019. 5.1, 5.2.3, 5.5.3, 5.6, 5.6.2

[DMS17] Amir Dembo, Andrea Montanari, and Subhabrata Sen. Extremal cuts of sparse random graphs. *The Annals of Probability*, 45(2):1190–1217, 2017. 5.1

[Dod84] Jozef Dodziuk. Difference equations, isoperimetric inequality and transience of certain random walks. *Trans. Amer. Math. Soc.*, 284(2):787–794, 1984. doi:10.2307/1999107. 1.1

[DP93] Charles Delorme and Svatopluk Poljak. Laplacian eigenvalues and the maximum cut problem. *Mathematical Programming*, 62(1–3):557–574, 1993. 5.1, 5.2.1, 5.2.4, 8

[DS16] Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limitations on learning DNF's. In *Proceedings of the 29th Annual Conference on Learning Theory*, pages 815–830, 2016. 5.1

[DSS15] Jian Ding, Allan Sly, and Nike Sun. Proof of the satisfiability conjecture for large k. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*, pages 59–68, 2015. 5.1

[EKZ20] Shai Evra, Tali Kaufman, and Gilles Zémor. Decodable quantum LDPC codes beyond the square root distance barrier using high dimensional expanders. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science*, pages 218–227. IEEE, 2020. arXiv:2004.07935, doi:10.1109/FOCS46700.2020.00029. 7.1

[Elo09] Yehonatan Elon. Gaussian waves on the regular tree. Technical Report 0907.5065, arXiv, 2009. 5.1

[ES63] Paul Erdős and Horst Sachs. Reguläre graphen gegebener tailenweite mit minimaler knollenzahl. *Wiss. Z. Univ. Halle-Willenberg Math. Nat.*, 12:251–258, 1963. 6.1

[Fei02] Uriel Feige. Relations between average case complexity and approximation complexity. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 543–543, 2002. 5.1, 5.1

[FK81] Z. Füredi and J. Komlós. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3):233–241, 1981. doi:10.1007/BF02579329. 2.2, 3.2, 4.1.2

[Fri93] Joel Friedman. Some geometric aspects of graphs and their eigenfunctions. *Duke Mathematical Journal*, 69(3):487–525, 1993. 1.1

[Fri08] Joel Friedman. A proof of Alon's second eigenvalue conjecture and related problems. *Memoirs of the American Mathematical Society*, 195(910):viii+100, 2008. 1.1.2, 1.1.16, 4.3, 5.1, 5.1.2

[Fru39] R. Frucht. Herstellung von Graphen mit vorgegebener abstrakter Gruppe. *Compositio Math.*, 6:239–250, 1939. URL: http://www.numdam.org/item?id=CM_1939__6__239_0. 7.1

[FS02] Uriel Feige and Gideon Schechtman. On the optimality of the random hyperplane

rounding technique for Max-Cut. *Randoom Structures and Algorithms*, 20(3):403–440, 2002. 5.2.1

[Gal62] R. G. Gallager. Low-density parity-check codes. *IRE Trans.*, IT-8:21–28, 1962. doi:10.1109/tit.1962.1057683. 1.1.3, 3.5, 6.1, 7.1

[GG81] Ofer Gabber and Zvi Galil. Explicit constructions of linear-sized superconcentrators. *Journal of Computer and System Sciences*, 22(3):407–420, 1981. Special issued dedicated to Michael Machtey. 1.1.1

[GLS84] M. Grötschel, L. Lovász, and A. Schrijver. Corrigendum to our paper: "The ellipsoid method and its consequences in combinatorial optimization" [Combinatorica **1** (1981), no. 2, 169–197; MR0625550 (84a:90044)]. *Combinatorica*, 4(4):291–295, 1984. doi:10.1007/BF02579139. 1

[GLS88] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer–Verlag, 1988. 8

[GLS93] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, second edition, 1993. doi:10.1007/978-3-642-78240-4. 1

[Gol00] Oded Goldreich. Candidate one-way functions based on expander graphs. Technical Report 90, Electronic Colloquium on Computational Complexity, 2000. 1.1.3, 5.1

[Gro53] Alexander Grothendieck. Résumé de la théorie métrique des produits tensoriels topologiques. *Boletín de la Sociedad Matemática São Paulo*, 8:1–79, 1953. 5.2.1

[GRS12] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. Essential coding theory. *Draft available at http://www. cse. buffalo. edu/atri/courses/coding-theory/book*, 2, 2012. 3.5

[GW95] Michel Goemans and David Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995. 5.1, 5.2.1

[GW21] Oded Goldreich and Avi Wigderson. Robustly self-ordered graphs: Constructions and applications to property testing. In Valentine Kabanets, editor, *Proceedings of the 36th Annual Computational Complexity Conference*, volume 200 of *LIPIcs*, pages 12:1–12:74. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.CCC.2021.12. 7.1

[GZ99] Rostislav Grigorchuk and Andrzej Żuk. On the asymptotic spectrum of random walks on infinite families of graphs. In *Random walks and discrete potential theory (Cortona, 1997)*, Sympos. Math., XXXIX, pages 188–204. Cambridge Univ. Press, Cambridge, 1999. 5.1.2

[Hås84] Johan Håstad. An NP-complete problem – some aspects of its solution and some possible applications. Master's thesis, Uppsala University, 1984. 4.1.1

[Has89] Ki-ichiro Hashimoto. Zeta functions of finite graphs and representations of p-adic

groups. In *Automorphic forms and geometry of arithmetic varieties*, volume 15 of *Advanced Studies in Pure Mathematics*, pages 211–280. Elsevier, 1989. 3.2.1, 3.2

[HHO21] Matthew B. Hastings, Jeongwan Haah, and Ryan O'Donnell. Fiber bundle codes: breaking the $n^{1/2}\mathrm{polylog}(n)$ barrier for quantum LDPC codes. In *Proceedings of the 53th Annual ACM Symposium on Theory of Computing*, pages 1276–1288. ACM, 2021. arXiv:2009.03921, doi:10.1145/3406325.3451005. 7.1, 7.1.2, 7.6

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *American Mathematical Society Bulletin*, 43(4):439–561, 2006. 1.1.1, 1.1.3, 8

[HPS18] Chris Hall, Doron Puder, and William F. Sawin. Ramanujan coverings of graphs. *Adv. Math.*, 323:367–410, 2018. doi:10.1016/j.aim.2017.10.042. 8

[HV15] Viktor Harangi and Bálint Virág. Independence ratio and random eigenvectors in transitive graphs. *The Annals of Probability*, 43(5):2810–2840, 2015. 5.1

[Iha66] Yasutaka Ihara. On discrete subgroups of the two by two projective linear group over $\mathfrak{p}$-adic fields. *Journal of the Mathematical Society of Japan*, 18:219–235, 1966. 1.1.1, 1.1, 3.2

[JŁR00] Svante Janson, Tomasz Łuczak, and Andrzej Rucinski. *Random graphs*. John Wiley & Sons, 2000. 4.4.1, 4.4.3

[JM21] Akhil Jalan and Dana Moshkovitz. Near-optimal cayley expanders for abelian groups. In *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, 2021. 7.3, 7.5.4

[JMO+22] Fernando Granha Jeronimo, Tushant Mittal, Ryan O'Donnell, Pedro Paredes, and Madhur Tulsiani. Explicit abelian lifts and quantum ldpc codes. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022. 1.2, 7

[JP00] Ari Juels and Marcus Peinado. Hiding cliques for cryptographic security. *Designs, Codes and Cryptography*, 20(3):269–280, 2000. 1.1.3, 5.1

[Kah95] Nabil Kahale. Eigenvalues and expansion of regular graphs. *Journal of the ACM (JACM)*, 42(5):1091–1106, 1995. 6.1.1, 6.2.2, 8

[Kas07] Martin Kassabov. Symmetric groups and expander graphs. *Inventiones Mathematicae*, 170(2):327–354, 2007. 3.4, 3.4.4, 3.4.6

[KK22] Amitay Kamber and Tali Kaufman. Combinatorics via closed orbits: number theoretic ramanujan graphs are not unique neighbor expanders. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 426–435, 2022. 8

[KMM+13] Florent Krzakala, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborová, and Pan Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences of the United States of America*, 110(52):20935–20940, 2013. 5.1.2

[KMOW17] Pravesh Kothari, Ryuhei Mori, Ryan O'Donnell, and David Witmer. Sum of squares lower bounds for refuting any CSP. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing*, pages 132–145, 2017. 5.1, 5.1

[KNR09] Eyal Kaplan, Moni Naor, and Omer Reingold. Derandomized constructions of k-wise (almost) independent permutations. *Algorithmica. An International Journal in Computer Science*, 55(1):113–133, 2009. 3.4, 3.4.4, 3.4.6

[Kow19] Emmanuel Kowalski. *An introduction to expander graphs*. Société mathématique de France, 2019. 1.1.3

[Kub12] Carlos S Kubrusly. *Spectral theory of operators on Hilbert spaces*. Springer Science & Business Media, 2012. 5.2.6, 5.2.25

[KW10] Tali Kaufman and Avi Wigderson. Symmetric LDPC codes and local testing. In Andrew Chi-Chih Yao, editor, *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 406–421. Tsinghua University Press, 2010. URL: `http://conference.iiis.tsinghua.edu.cn/ICS2010/content/papers/32.html`. 7.1

[LBM+18] Huaan Li, Baoming Bai, Xijin Mu, Ji Zhang, and Hengzhou Xu. Algebra-assisted construction of quasi-cyclic LDPC codes for 5G new radio. *IEEE Access*, 6:50229–50244, 2018. `doi:10.1109/ACCESS.2018.2868963`. 7.1

[Lin17] Huijia Lin. Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs. In *Proceedings of the 37th Annual International Cryptography Conference*, pages 599–629, 2017. 5.1

[Lov79] László Lovász. On the Shannon capacity of a graph. *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, 25(1):1–7, 1979. 5.2.1

[LPS88] Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261—-277, 1988. 1.1.1, 1.1.1, 1.1, 6.1, 7.1.2

[LR00] John Lafferty and Dan Rockmore. Codes and iterative decoding on algebraic expander graphs. In *the Proceedings of ISITA*. Citeseer, 2000. 6.1

[LS21] Nati Linial and Michael Simkin. A randomized construction of high girth regular graphs. *Random Structures Algorithms*, 58(2):345–369, 2021. `doi:10.1002/rsa.20976`. 6.1

[LU95] Felix Lazebnik and Vasiliy A. Ustimenko. Explicit construction of graphs with an arbitrary large girth and of large size. *Discrete Appl. Math.*, 60(1-3):275–284, 1995. ARIDAM VI and VII (New Brunswick, NJ, 1991/1992). `doi:10.1016/0166-218X(94)00058-L`. 6.1

[Lyo17] Russell Lyons. Factors of IID on trees. *Combinatorics, Probability and Computing*, 26(2):285–300, 2017. 5.1

[Mar73] Grigory Margulis. Explicit construction of concentrators. *Problemy Peredachi Informatsii*, 94(4):71–80, 1973. 1.1.1

[Mar82] G. A. Margulis. Explicit constructions of graphs without short cycles and low den-

sity codes. *Combinatorica*, 2(1):71–78, 1982. doi:10.1007/BF02579283. 6.1

[Mar88] Grigory Margulis. Explicit group-theoretic constructions of combinatorial schemes and their applications in the construction of expanders and concentrators. *Problemy Peredachi Informatsii*, 24(1):51–60, 1988. 1.1.1, 1.1, 6.1

[Mas14] Laurent Massoulié. Community detection thresholds and the weak ramanujan property. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 694–703. ACM, 2014. 5.1.2

[MM09] Marc Mézard and Andrea Montanari. *Information, physics, and computation*. Oxford University Press, 2009. 5.1

[MM21] Theo McKenzie and Sidhanth Mohanty. High-girth near-Ramanujan graphs with lossy vertex expansion. In *48th International Colloquium on Automata, Languages, and Programming*, volume 198 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 96, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2021. 8

[MNS18] Elchanan Mossel, Joe Neeman, and Allan Sly. A proof of the block model threshold conjecture. *Combinatorica*, 38(3):665–708, 2018. 5.1.2

[MO18] Sidhanth Mohanty and Ryan O'Donnell. X-Ramanujan graphs, 2018. Available at https://arxiv.org/abs/1904.03500. 5.1.2, 5.1.2, 5.2.4

[Mon17] Andrea Montanari. Bounds on ground state enery in the Sherrington–Kirkpatrick model, 2017. Open problem from AIM workshop, available at http://aimpl.org/phaserandom/1/. 5.1

[Mon19] Andrea Montanari. Optimization of the Sherrington-Kirkpatrick Hamiltonian. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science*, pages 1417–1433. IEEE Comput. Soc. Press, Los Alamitos, CA, 2019. doi:10.1109/FOCS.2019.00087. 5.1

[MOP20a] Sidhanth Mohanty, Ryan O'Donnell, and Pedro Paredes. Explicit near-ramanujan graphs of every degree. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 510–523, 2020. 1.2, 4, 6.1.1, 8

[MOP20b] Sidhanth Mohanty, Ryan O'Donnell, and Pedro Paredes. The sdp value for random two-eigenvalue csps. In *37th International Symposium on Theoretical Aspects of Computer Science (STACS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. 1.2, 5

[Mor94] Moshe Morgenstern. Existence and explicit constructions of $q+1$ regular Ramanujan graphs for every prime power q. *Journal of Combinatorial Theory. Series B*, 62(1):44–62, 1994. 1.1.1, 1.1.10, 1.1, 6.1, 8

[MPR16] Raffaele Marino, Giorgio Parisi, and Federico Ricci-Tersenghi. The backtracking survey propagation algorithm for solving random k-sat problems. *Nature Communications*, 7:12996, 2016. 5.1

[MS02] Mohammad M Mansour and Naresh R Shanbhag. Construction of ldpc codes from ramanujan graphs. In *36th Annu. Conf. on Information Sciences and Systems*,

2002. 6.1

[MS16] Andrea Montanari and Subhabrata Sen. Semidefinite programs on sparse random graphs and their application to community detection. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing*, pages 814–827, 2016. 5.1

[MSS15a] Adam Marcus, Daniel Spielman, and Nikhil Srivastava. Interlacing families I: Bipartite Ramanujan graphs of all degrees. *Annals of Mathematics. Second Series*, 182(1):307–325, 2015. 1.1.1, 1.1.15, 1.1, 7.1, 8

[MSS15b] Adam Marcus, Daniel Spielman, and Nikhil Srivastava. Interlacing families IV: Bipartite Ramanujan graphs of all sizes. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, pages 1358–1377, 2015. 1.1.1, 1.1.15, 1.1

[MWW04] Brendan D. McKay, Nicholas C. Wormald, and Beata Wysocka. Short cycles in random regular graphs. *Electron. J. Combin.*, 11(1):Research Paper 66, 12, 2004. URL: `http://www.combinatorics.org/Volume_11/Abstracts/v11i1r66.html`. 6.1, 6.3.2, 6.4.1

[Nil91] A. Nilli. On the second eigenvalue of a graph. *Discrete Mathematics*, 91(2):207–210, 1991. 1.1

[NN93] Joseph Naor and Moni Naor. Small-bias probability spaces: efficient constructions and applications. *SIAM Journal on Computing*, 22(4):838–856, 1993. 3.4, 3.4.2

[Nor97] Sam Northshield. Several Proofs of Ihara's theorem. 1997. 5.3

[OST+14] Ryan O'Donnell, Xiaorui Sun, Li-Yang Tan, John Wright, and Yu Zhao. A composition theorem for parity kill number. In *Proceedings of the 29th Annual Computational Complexity Conference*, pages 144–154, 2014. 5.2.2

[Par21] Pedro Paredes. Spectrum preserving short cycle removal on regular graphs. In *38th International Symposium on Theoretical Aspects of Computer Science*, 2021. 1.2, 6

[Piz90] Arnold Pizer. Ramanujan graphs and Hecke operators. *American Mathematical Society. Bulletin. New Series*, 23(1):127–137, 1990. 1.1.1, 1.1

[PK21] Pavel Panteleev and Gleb Kalachev. Quantum LDPC Codes with Almost Linear Minimum Distance. *IEEE Transactions on Information Theory*, December 2021. `arXiv:2012.04068`, `doi:10.1109/TIT.2021.3119384`. 7.1, 7.1.2, 7.1.4, 7.1.2, 7.1.5, 7.6, 7.6.1, 7.6, 7.6.2, 7.6

[PK22] Pavel Panteleev and Gleb Kalachev. Asymptotically good quantum and locally testable classical ldpc codes. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 375–388, 2022. 7.1, 8

[Ram15] Farzaneh Ramezani. On the signed graphs with two distinct eigenvalues. *arXiv preprint arXiv:1511.03511*, 2015. 5.2.3

[RRS17] Prasad Raghavendra, Satish Rao, and Tselil Schramm. Strongly refuting random CSPs below the spectral threshold. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing*, pages 121–131, 2017. 5.1

[RU08] Tom Richardson and Ruediger Urbanke. *Modern coding theory*. Cambridge university press, 2008. doi:10.1017/CBO9780511791338. 3.5, 7.1

[RV00] Joachim Rosenthal and Pascal O Vontobel. Constructions of ldpc codes using ramanujan graphs and ideas from margulis. In *in Proc. of the 38-th Allerton Conference on Communication, Control, and Computing*. Citeseer, 2000. 6.1

[RVW02] Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders. *Annals of Mathematics*, 155(1):157–187, 2002. 1.1.1, 1.1.11, 1.1, 8, 8

[Sen18] Subhabrata Sen. Optimization on sparse random hypergraphs and spin glasses. *Random Structures & Algorithms*, 53(3):504–536, 2018. 5.1

[Ser77] Jean-Pierre Serre. *Arbres, amalgames,* SL_2. Société Mathématique de France, Paris, 1977. Avec un sommaire anglais, Rédigé avec la collaboration de Hyman Bass, Astérisque, No. 46. 3.2

[Sho90] Victor Shoup. New algorithms for finding irreducible polynomials over finite fields. *Mathematics of Computation*, 54(189):435–447, 1990. 3.4

[Spi19] Daniel Spielman. *Spectral and Algebraic Graph Theory*. Incomplete Draft, December 2019. 3.1, 3.1

[SS96] Michael Sipser and Daniel A. Spielman. Expander codes. volume 42, pages 1710–1722. 1996. Codes and complexity. doi:10.1109/18.556667. 1.1.3

[Ste96] Andrew Steane. Multiple-particle interference and quantum error correction. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 452(1954):2551–2577, Nov 1996. arXiv:quant-ph/9601029v3, doi:10.1098/rspa.1996.0136. 7.1

[Tal06] Michel Talagrand. The Parisi formula. *Annals of Mathematics. Second Series*, 163(1):221–263, 2006. 5.1

[Tsi80] Boris Tsirelson. Quantum generalizations of Bell's inequality. *Letters in Mathematical Physics*, 4(2):93–100, 1980. 5.2.6, 5.2.7

[WF09] Yusuke Watanabe and Kenji Fukumizu. Graph zeta function in the Bethe free energy and loopy belief propagation. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems*, pages 2017–2025, 2009. 3.2, 3.2.3

[Wil07] Ryan Williams. *Algorithms and resource requirements for fundamental problems*. PhD thesis, Carnegie Mellon University, 2007. 5.2.2

[Wor99] Nicholas Wormald. Models of random regular graphs. In *Surveys in combinatorics, 1999 (Canterbury)*, volume 267 of *London Mathematical Society Lecture Note Series*, pages 239–298. Cambridge Univ. Press, Cambridge, 1999. 3.3, 4.4.1