

The SDP value for random two-eigenvalue CSPs

Sidhanth Mohanty* Ryan O’Donnell† Pedro Paredes†

June 16, 2019

Abstract

We precisely determine the SDP value (equivalently, quantum value) of large random instances of certain kinds of constraint satisfaction problems, “two-eigenvalue 2CSPs”. We show this SDP value coincides with the spectral relaxation value, possibly indicating a computational threshold. Our analysis extends the previously resolved cases of random regular 2XOR and NAE-3SAT, and includes new cases such as random Sort_4 (equivalently, CHSH) and Forrelation CSPs. Our techniques include new generalizations of the nonbacktracking operator, the Ihara–Bass Formula, and the Friedman/Bordenave proof of Alon’s Conjecture.

*EECS Department, University of California Berkeley. Supported by NSF grant CCF-1718695

†Computer Science Department, Carnegie Mellon University. Supported by NSF grant CCF-1717606. This material is based upon work supported by the National Science Foundation under grant numbers listed above. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation (NSF).

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Our results | 3 |
| 1.2 | Sketch of our techniques | 4 |
| 1.2.1 | Friedman/Bordenave Theorems for two-eigenvalue additive lifts | 6 |
| 1.3 | Relationship to the work of Bordenave–Collins | 8 |
| 2 | Preliminaries | 9 |
| 2.1 | 2XOR optimization problems and their relaxations | 9 |
| 2.2 | Quantum games, and some quantum-relevant constraints | 11 |
| 2.3 | 2XOR graphs with only 2 distinct eigenvalues | 12 |
| 2.4 | Random constraint graphs, instance graphs, and additive products | 13 |
| 2.5 | Nomadic walks operators | 15 |
| 2.6 | Operator Theory | 16 |
| 3 | An Ihara–Bass formula for additive lifts of 2-eigenvalue atoms | 17 |
| 4 | Connecting the adjacency and nomadic spectrum | 22 |
| 5 | Additive products of 2-eigenvalue atoms | 25 |
| 5.1 | Enclosing the spectrum | 26 |
| 5.2 | Construction of Witness Vectors | 28 |
| 5.3 | SDP solution for random additive lifts | 31 |
| 6 | Friedman/Bordenave for additive lifts | 33 |
| 6.1 | Trace Method setup, and getting rid of tangles | 34 |
| 6.2 | Eliminating singletons, and reduction to counting | 36 |
| 6.3 | Tangle-free, singleton-free linkages are nearly duplicative | 38 |
| 6.4 | The final countdown | 40 |
| 7 | The SDP value for random two-eigenvalue CSPs | 45 |

1 Introduction

This work is concerned with the average-case complexity of constraint satisfaction problems (CSPs). In the theory of algorithms and complexity, the most difficult instances of a given 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 k SAT 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 latter 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 $\text{OPT} \approx \frac{1}{2} + \frac{P^*}{\sqrt{d}}$. 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 $-\lambda_{\min}(G) \leq 2\sqrt{d-1} + o_n(1)$ whp; thus computing the smallest eigenvalue efficiently certifies $\text{OPT} \lesssim \frac{1}{2} + \frac{1}{\sqrt{d}}$. 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

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

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 [Mon18] has recently given an efficient PTAS for the search problem², 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 k SAT or k XOR, $k \geq 3$ — 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 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⁺19] 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 $c \geq 8$. 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 $c \geq 14$ even the basic spectral algorithm certifies unsatisfiability, whereas for $c \leq 13$ even the SDP augmented with “triangle inequalities”

²Modulo a widely believed analytic assumption.

³We 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.

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 $c \geq 3$, 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.

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 1.2.1](#) 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 Sort₄ (a.k.a. CHSH) predicate; and, 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 2.2](#).

We state here two theorems that our new techniques allow us to prove. Recall the Sort₄ predicate, which is satisfied iff its 4 Boolean inputs x_1, x_2, x_3, x_4 satisfy $x_1 \leq x_2 \leq x_3 \leq x_4$ or $x_1 \geq x_2 \geq x_3 \geq x_4$. We precisely define “random c -constraint-regular CSP instance” in [Section 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 1.1. *For random c -constraint-regular instances of the Sort₄-CSP, the SDP-satisfiability threshold occurs (in a sense) at $c = 4 + 2\sqrt{2} \approx 6.83$. Indeed, if $c \geq 7$ then even the basic eigenvalue bound certifies unsatisfiability (whp); and, if $c \leq 6$ 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 Sort₄-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.

⁴Our 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.

Generalizing this, the Forrelation_k constraint is a certain (quantum-inspired) map $\{\pm 1\}^{2^k+2^k} \rightarrow [-1, +1]$ 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 2.2](#); here we just additionally remark that Forrelation_1 corresponds to the “CHSH game”, and that $\frac{1}{2} + \text{Forrelation}_1$ is equivalent to the Sort_4 predicate.

Theorem 1.2. *For random c -constraint-regular instances of the Forrelation_k -CSP and any constant $\varepsilon > 0$, the SDP value is whp in the range $\frac{2\sqrt{c-1}}{c^{2k/2}} \pm \varepsilon$. This is also true of the eigenvalue bound.*

When considering the SDP value for $\frac{1}{2} + \text{Forrelation}_1$, the formula above crosses the threshold of 1 when $c = 4 + 2\sqrt{2}$, yielding the statement in [Theorem 1.1](#) about the SDP-satisfiability threshold of random c -constraint-regular Sort_4 -CSPs.

1.2 Sketch of our techniques

Here we sketch how our results like [Theorem 1.1](#) and [Theorem 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:

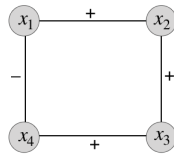


Figure 1: The Sort_4 predicate

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

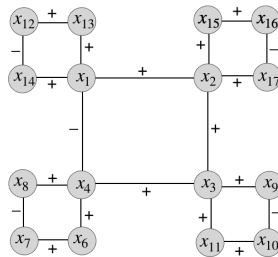


Figure 2: Piece of Sort_4 instance

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, \dots, x_n so as to maximize $\frac{1}{n} \sum_{ij} A_{ij} x_i x_j$, where $A \in$

⁵In 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.

$\{0, \pm 1\}^{n \times n}$ is the adjacency matrix of the edge-signed graph partially depicted above. The “eigenvalue upper bound” $\text{EIG}(\mathcal{I})$ 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” $\text{SDP}(\mathcal{I})$ (which is at least as tight: $\text{SDP}(\mathcal{I}) \leq \text{EIG}(\mathcal{I})$) 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

$$\text{EIG}(\mathcal{I}) \lesssim f(c) \lesssim \text{SDP}(\mathcal{I}) \tag{1}$$

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 $\text{SDP}(\mathcal{I}) \leq \text{EIG}(\mathcal{I})$ always.

In this section we mainly describe how to obtain the optimal inequality on the left in (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 [Bor15]. 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:

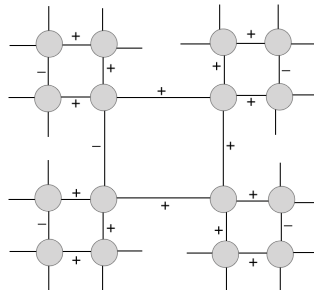


Figure 3: Sort_4 infinite graph

Here $X := \text{Sort}_4 \ast \text{Sort}_4$ is the so-called *additive product* of 2 copies of the Sort_4 graph, a notion recently introduced in [MO18]. 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 (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.)

⁶More 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.

1.2.1 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 $\rho(X) - o_n(1)$ 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 *oversimplifies* 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 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 rearranging the

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

variables in the Sort_4 predicate, its adjacency matrix is

$$A = \begin{pmatrix} 0 & 0 & +1 & +1 \\ 0 & 0 & +1 & -1 \\ +1 & +1 & 0 & 0 \\ +1 & -1 & 0 & 0 \end{pmatrix}, \quad (2)$$

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 2.4](#) for a formal definition):

Definition 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 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 1.4](#) in [Section 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 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 6](#):

Theorem 1.5. *With high probability,*

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

And we can use our version of Ihara–Bass, [Theorem 1.4](#), to conclude bounds on the spectrum of the adjacency matrix A from [Theorem 1.5](#), which is worked out in [Section 4](#).

Theorem 1.6. *With high probability,*

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

Yet another advantage of using nomadic walks instead of closed walks is that in [Theorem 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 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](#).

Theorem 1.7. *For every $\varepsilon > 0$, whp there exists a PSD matrix M with an all-ones diagonal such that*

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

As detailed out in [Section 7](#), this lets us conclude the main theorem of this paper:

Theorem 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 1.2](#) can be viewed as a special case of [Theorem 1.8](#).

1.3 Relationship to the work of Bordenave–Collins

Xinyu Wu has brought to our attention the relevance to our work of a recent paper by Bordenave and Collins [[BC18](#)]. Briefly put, their paper establishes a Friedman/Bordenave theorem for large random graphs whose adjacency matrices are noncommutative polynomials in a fixed number of independent random matching matrices and permutation matrices (together with their transposes). As a most basic example, it recovers the following form of Friedman’s Theorem: whp, the sum of d random perfect matchings has all nontrivial eigenvalues bounded in magnitude by $\rho(\mathbb{Z}_2 * \cdots (d \text{ times}) \cdots * \mathbb{Z}_2) + o_n(1) = 2\sqrt{d-1} + o_n(1)$. However, the Bordenave–Collins work gives much more than this. For example, let G be the n -vertex graph formed as

$$P + P^\top + M - PMP^\top,$$

where M is a random matching matrix and P is an independent random permutation matrix. It is not hard to see that G will essentially “locally resemble” a 2-constraint-regular Sort_4 -CSP instance. And, the Bordenave–Collins work implies that the eigenvalues of G are bounded (whp) by $\rho(\text{Sort}_4 \blacklozenge \text{Sort}_4)$. Using the theory of free probability, it is possible to directly compute that $\rho(\text{Sort}_4 \blacklozenge \text{Sort}_4) = 2\sqrt{2}$. In this way, our [Theorem 1.6](#) in the case of 2-constraint-regular Sort_4 -CSPs is covered by Bordenave and Collins. Indeed, it is not hard to generalize this example to the case of c -constraint-regular Sort_4 -CSPs for any *even* integer c .

Indeed, the Bordenave–Collins work also treats some kinds of graphs that our work cannot; for example, Wu gave the example when G is the n -vertex graph generated by the polynomial

$$P_1 + P_1^\top + P_2 + P_2^\top + P_3 + P_3^\top + P_4 + P_4^\top + P_1 P_2 P_3 P_4 + P_4^\top P_3^\top P_2^\top P_1^\top,$$

where P_1, \dots, P_4 are independent uniformly random permutation matrices. This G “locally resembles” the infinite free product graph $X = \mathbb{Z}_4 * \mathbb{Z}_4 * \mathbb{Z}_4 * \mathbb{Z}_4$, and the Bordenave–Collins work implies that whp, G ’s nontrivial eigenvalues are bounded in magnitude by $\rho(X) + o_n(1)$. (We remark that computing the numeric value of this $\rho(X)$ is difficult, but possible; see, e.g., [Woe00, Ch. 9C]). Since the 4-cycle graph \mathbb{Z}_4 has more than two distinct eigenvalues, it is not covered by our work.

This said, the Bordenave–Collins work does not subsume our [Theorem 1.6](#), as there are plenty of graph families that our theorem handles but Bordenave–Collins’s does not (seem to). For example, Wu has sketched to us a proof that one cannot obtain c -constraint-regular Sort_4 instances for *odd* c through any straightforward use of [BC18]. Additionally, even in the cases of interest to us where Bordenave–Collins applies, we can point to some (minor) advantages of our methods. For one, our model of random graph generation clearly corresponds to precisely-regular CSP instances, whereas in the Bordenave–Collins model there will be (in expectation) a constant number of local “blemishes” where one cannot interpret a piece of the graph as a constraint. For another, our work directly yields the numerical values of the appropriate spectral radii $\rho(X)$ (though in the cases where our results apply, these can be obtained through standard methods in free probability).

2 Preliminaries

2.1 2XOR optimization problems and their relaxations

All of the CSPs studied in this work (Max-Cut, NAE-3SAT, Sort_4 , 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 2.1. (Optimization of 2XOR instances) Let $G = (V, E)$ be an undirected graph (possibly with parallel edges), with edge-signing $\text{wt} : E \rightarrow \{\pm 1\}$. We call the pair $\mathcal{I} = (G, \text{wt})$ an *instance*. The associated 2XOR *optimization problem* is to determine the (*true*) *optimum value*

$$\text{OPT}(\mathcal{I}) = \max_{x: V \rightarrow \{\pm 1\}} \text{avg}_{e=\{u,v\} \in E} \{\text{wt}(e)x_u x_v\} \in [-1, +1].$$

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

Determining $\text{OPT}(\mathcal{I})$ 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 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 $\text{wt}(e)$ 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:

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

Definition 2.3. (Eigenvalue bound) The *eigenvalue bound* $\text{EIG}(\mathcal{I})$ for 2XOR instance \mathcal{I} with adjacency matrix A is $\frac{n}{2|E|} \lambda_{\max}(A)$, where λ_{\max} denotes the maximum eigenvalue. We have $\text{OPT}(\mathcal{I}) \leq \text{EIG}(\mathcal{I})$ always, as the eigenvalue bound captures the relaxation of 2XOR optimization where we allow any $x : V \rightarrow \mathbb{R}$ satisfying $\|x\|^2 = n$.

The *SDP value* provides an even tighter upper bound on $\text{OPT}(\mathcal{I})$, 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 2.4. (SDP bound) The *SDP bound* $\text{SDP}(\mathcal{I})$ for 2XOR instance \mathcal{I} is

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

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} :

$$\text{OPT}(\mathcal{I}) \leq \text{SDP}(\mathcal{I}) \leq \text{EIG}(\mathcal{I}).$$

The left inequality is obvious. One way to see the right inequality is to use the fact [DP93], based on SDP duality, that $\text{SDP}(\mathcal{I})$ 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

$$\frac{1}{2} + \frac{1}{2} \text{SDP}(\mathcal{I}) \leq 1.138 \left(\frac{1}{2} + \frac{1}{2} \text{OPT}(\mathcal{I}) \right)$$

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

- ([CW04]) $\text{SDP}(\mathcal{I}) \leq O(\text{OPT}(\mathcal{I}) \cdot \log(1/\text{OPT}(\mathcal{I})))$ always holds.
- When G is bipartite (a special case of particular interest, see Section 2.2), it holds that $\text{SDP}(\mathcal{I}) \leq K \cdot \text{OPT}(\mathcal{I})$ for constant K . This is known as *Grothendieck’s inequality* [Gro53], and the constant is known [BMMN13] to satisfy $K < \pi / (2 \ln(1 + \sqrt{2})) \approx 1.78$.

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

⁹The 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 $\{\pm 1\}^5$ — already has $\text{OPT} = 3/5$ and $\text{SDP} = (1 + \sqrt{5})/4$, showing that 1.138 cannot be improved below 1.131.

2.2 Quantum games, and some quantum-relevant constraints

In the case when the underlying graph G is bipartite, $\text{SDP}(\mathcal{I})$ 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 Forrelation_k constraints from [Theorem 1.1](#) and [Theorem 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 Forrelation_k .

Let us now recall the relevant quantum facts.

Definition 2.5 (Nonlocal 2XOR games). Given a 2XOR instance $\mathcal{I} = (G, \text{wt})$ 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 $\text{wt}(e)x_u y_v$. 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 $\text{OPT}(\mathcal{I})$.

Theorem 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 $\text{SDP}(\mathcal{I})$.*

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

Example 2.7. Consider the 2XOR instance depicted in [Figure 4](#), called CHSH after Clauser, Horne, Shimony, and Holt [\[CHSH69\]](#). It has

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

The upper bound $4 \cdot \text{OPT}(\text{CHSH}) \leq 2$ is often called *Bell’s inequality* [\[Bel64\]](#), and the higher lower

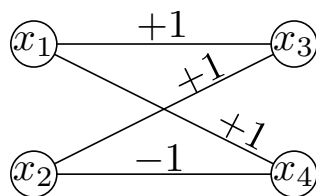


Figure 4: The CHSH game/CSP

bound $1/\sqrt{2} \leq \text{SDP}(\text{CHSH})$ is from [\[CHSH69\]](#) (with $\text{SDP}(\text{CHSH}) \leq 1/\sqrt{2}$ 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. [\(2\)](#)),

$$\text{CHSH}(x_1, x_2, x_3, x_4) = \frac{1}{4}(x_1 x_3 + x_2 x_3 + x_1 x_4 - x_2 x_4) = \text{Sort}_4(x_2, x_3, x_1, x_4) - \frac{1}{2}.$$

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 Forrelation_k function is defined by

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

where $H_k = \begin{pmatrix} +1 & +1 \\ +1 & -1 \end{pmatrix}^{\otimes k}$ is the k th Walsh–Hadamard matrix. Note that Forrelation_0 corresponds to the single-(positive-)edge 2XOR CSP, and Forrelation_1 is CHSH.

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 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⁺19]. This is because

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

Let us make some definitions we will use throughout the paper.

Definition 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 2.8 and prove some properties that will be used throughout the paper.

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

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

2.4 Random constraint graphs, instance graphs, and additive products

Definition 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 *1st variable group*, the *2nd 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 *1st constraint/atom group*, *2nd 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 i th group has exactly one neighbor from each of the c constraint groups, and similarly every constraint vertex in the j th group has exactly one neighbor from each of the r variable groups. An example of a constraint graph is shown in [Figure 6](#).¹⁰

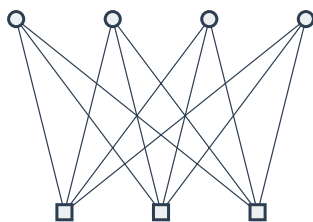


Figure 5: The complete $K_{4,3}$ graph

Definition 2.13 (Instance graphs). Let $\mathcal{A} = (A_1, \dots, A_c)$ be a sequence of *atoms*, meaning edge-weighted undirected graphs on a common vertex set $[r]$. (In this paper, 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* $\mathcal{I} := \mathcal{A}(\mathcal{H})$. 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 j th constraint group of \mathcal{H} . Given f , with variables neighbors v_1, \dots, 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, \dots, v_r as A_f , and for any edge e in \mathcal{I} that came from placing A_j , we define $\text{Atom}(e) := A_f$. We use $v \sim A_f$ to denote that v is one of v_1, \dots, v_r . For $u, v \in \{v_1, \dots, v_r\}$, $A_f(u, v)$ 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 $\text{Atoms}(\mathcal{I})$. An example of an instance graph and corresponding constraint graph is shown in [Figure 6](#).

Remark 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)

¹⁰This 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 $1 \dots r$ in an arbitrary way, fixing *their* constraint neighbors to be in groups $2 \dots c$ in an arbitrary way, etc.

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 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 3.1](#).

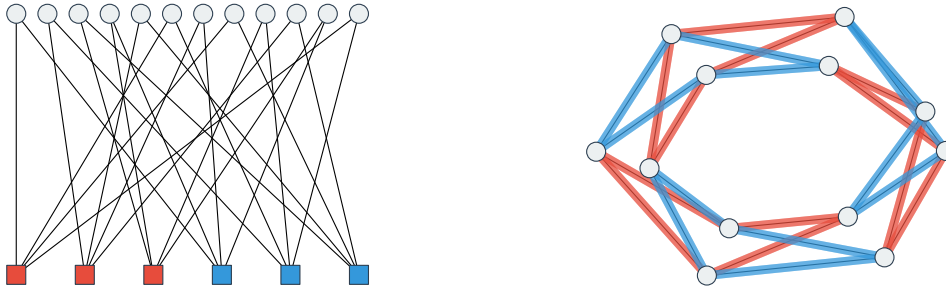


Figure 6: The figure on the left shows an example of a 4-ary, 2-atom 3-fold lift constraint graph, with the left bipartition color coded by constraint/atom groups. The figure on the right is the corresponding instance graph on (C_4, C_4) , two four-cycle graphs, where each atom is color coded to match the figure on the left.

The following notions of additive lifts and additive products were introduced in [\[MO18\]](#):

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

Definition 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, \dots, A_c , denoted $A_1 \diamond A_2 \diamond \dots \diamond A_c$.

We will also extend [Definition 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 ζ^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 ζ^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, \dots, v_r , we assign independent and uniformly random signs $(\zeta_i^f)_{i \in [r]}$ to the variables. Then when the copy of A_j is added into \mathcal{I} , the $\{i, i'\}$ -edge has its weight multiplied by $\zeta_i^f \zeta_{i'}^f$. 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 $(\zeta_{ij}^f)_{i,j \in [r]}$ which are “1-wise uniform”. This is the only property we will require of a negation model.

Definition 2.18 (Random additive lifts with negations). A random additive lift *with 1-wise uniform negations* is a variant of [Definition 2.13](#) where, for each constraint vertex f there are associated random signs $\zeta_i^{(f)} \in \{\pm 1\}$, where $i \in [r]$. For each fixed f , the random variables $\zeta_i^{(f)}$ are required to be ± 1 with probability $1/2$ each, but they may be arbitrarily correlated; across different f 's, the collections $(\zeta_i^{(f)})_{i \in [r]}$ 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 $\zeta_{ii'}^{(f)} := \zeta_i^{(f)} \zeta_{i'}^{(f)}$.

Remark 2.19. For a given constraint-vertex f of an instance graph \mathcal{I} obtained via a random additive lift with negations, the matrix $\text{Adj}(A_f)$ has the same spectrum as $\text{Adj}(\overline{A}_f)$ where \overline{A}_f denotes the subgraph prior to applying random negations, since there is a sign diagonal matrix D such that $\text{Adj}(\overline{A}_f) = D \cdot \text{Adj}(A_f) \cdot D^\dagger$.

2.5 Nomadic walks operators

Definition 2.20 (Nomadic walks). Let \mathcal{H} be a constraint graph, $\mathcal{A} = (A_1, \dots, A_c)$ 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.

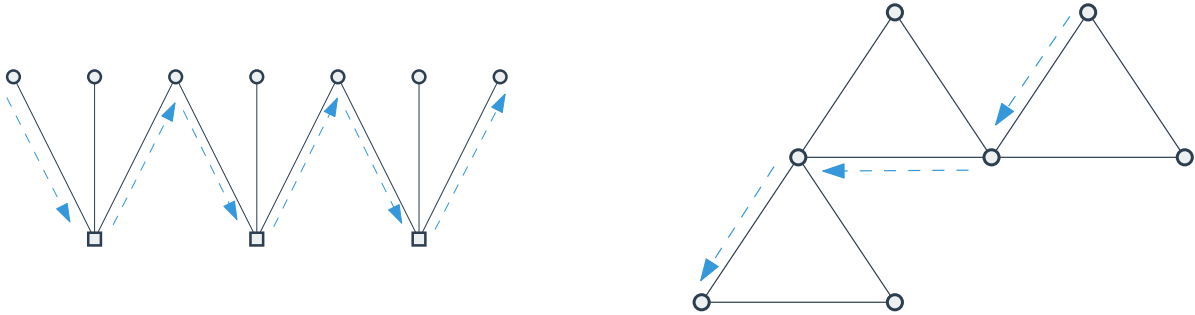


Figure 7: The figure on the left shows a nonbacktracking walk on a subset of a 3-ary constraint graph and the one on the right the same nomadic walk on the corresponding instance graph.

Definition 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., $\text{wt}(\vec{e}) = \text{wt}(\vec{e}^{-1}) = \text{wt}(e)$. Let \vec{E} denote the collection of all directed edges. Now B is defined to be the

following linear operator on $\ell_2(\vec{E})$:

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

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 $B[\vec{e}, \vec{e}'] = \text{wt}(\vec{e}')$ 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.)

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 $T : \ell_2(V) \rightarrow \ell_2(V)$ be a bounded, self-adjoint linear operator.

Definition 2.22. We refer to the *spectrum* of T , $\text{Spec}(T)$, as the set of all complex λ such that $\lambda \mathbb{1} - T$ is not invertible. $\text{Spec}(T)$ is a nonempty, compact set.

Definition 2.23. We call λ an *approximate eigenvalue* of T if for every $\varepsilon > 0$, there is unit x in \mathcal{X} such that $\|Tx - \lambda x\| \leq \varepsilon$. We call such an x an ε -*approximate eigenvector* or ε -*approximate eigenfunction*.

Theorem 2.24. *If T is a self-adjoint operator, then every $\lambda \in \text{Spec}(T)$ is an approximate eigenvalue.*

Theorem 2.25. [Consequence of Proposition 4.L of [Kub12]] *If λ is an isolated point in $\text{Spec}(T)$, then it is an eigenvalue of T , i.e., it is a 0-approximate eigenvalue.*

Corollary 2.26. $\lambda_{\min} := \min\{\text{Spec}(T)\}$ and $\lambda_{\max} := \max\{\text{Spec}(T)\}$ are both approximate eigenvalues of T .

Fact 2.27. *Additionally,*

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

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

Definition 2.28. The *spectral radius* $\rho(T)$ is defined as $\max_{\sigma \in \text{Spec}(T)} |\sigma|$.

Definition 2.29. The *operator norm* of T , denoted $\|T\|_{\text{op}}$, is defined as

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

Fact 2.30. $\rho(T) = \lim_{k \rightarrow \infty} \|T^k\|_{\text{op}}^{1/k}$.

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 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.

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

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

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

We use $\text{wt}(e)$ to denote the weight on edge e , and define the weight of a walk $W = e_1 e_2 \dots e_\ell$ as

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

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

Claim 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 $s \leq k - 1$, 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 i th 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 $\mathcal{W}_{uv}^{(1)}$.
2. $W_k = W_{k-1}^{-1}$. Call the collection of these walks $\mathcal{W}_{uv}^{(2)}$.
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 $\mathcal{W}_{uv}^{(3)}$.

Suppose $k \geq 3$.

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

We apply [Fact 2.10](#) and get

$$\begin{aligned}
&= \sum_{\substack{W' \text{ (} k-2 \text{)-length nomadic walk} \\ \text{from } u \text{ to } v}} \text{wt}(W') (c-1) (-\lambda_1 \lambda_2) \\
&= (c-1) (-\lambda_1 \lambda_2) p_{k-2}(A)_{uv}.
\end{aligned}$$

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

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

We do a similar calculation for $\mathcal{W}_{uv}^{(3)}$ for $k \geq 2$. Observe that W_{k-1} and W_k have to be in the same atom, which we denote $\text{Atom}(W_{k-1})$. Thus, there is an edge e^* between w_{k-2} and v in $\text{Atom}(W_{k-1})$ too (see [Remark 2.15](#)).

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

By applying [Fact 2.11](#), we get

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

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

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

$$\sum_{W \in \mathcal{W}_{uv}^{(1)}} \text{wt}(W) = Ap_{k-1}(A)_{uv} - ((c-1)(-\lambda_1\lambda_2)p_{k-2}(A)_{uv} + (\lambda_1 + \lambda_2)p_{k-1}(A)_{uv})$$

$$\sum_{W \in \mathcal{W}_{uv}^{(1)}} \text{wt}(W) = p_k(A)_{uv}.$$

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. \square

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

$$\frac{d}{dt} \log \det M(t) = \text{tr} \left(M'(t) M(t)^{-1} \right)$$

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

Corollary 3.4. *Taking $M(t) = \mathbb{1} - Ht$ for a fixed square matrix H yields*

$$\frac{d}{dt} \log \det(\mathbb{1} - Ht) = \text{tr} \left(-H(\mathbb{1} - Ht)^{-1} \right) \implies -t \frac{d}{dt} \log \det(\mathbb{1} - Ht) = \sum_{k \geq 1} \text{tr}(H^k) t^k.$$

Regarding this corollary, we can derive the statement about the power sums of the roots of a polynomial $f(t)$ by taking $H = \text{diag}(\lambda_1, \dots, \lambda_n)$ where the λ_i ’s are the roots of f . On the other hand, it actually suffices to prove [Corollary 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 $((c-1)(-\lambda_1\lambda_2))^t$ (see [Lemma 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 generate the polynomials $p_k(x)$, one can conclude

Lemma 3.5. $F(t) = AF(t)t - (\lambda_1 + \lambda_2)F(t)t - \alpha_{\text{gr}}F(t)t^2 + (1 + t\lambda_1)(1 + t\lambda_2)\mathbb{1}.$

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

Corollary 3.6. $(1 + \lambda_1 t)^{-1}(1 + \lambda_2 t)^{-1} \cdot (\mathbb{1} - At + (\lambda_1 + \lambda_2)t\mathbb{1} + \alpha_{\text{gr}}t^2\mathbb{1})F(t) = \mathbb{1}.$ In other words, $F(t) = (1 + \lambda_1 t)(1 + \lambda_2 t)\mathbb{1} \cdot L(t)^{-1}$, where $L(t) := \mathbb{1} - At + (\lambda_1 + \lambda_2)t\mathbb{1} + \alpha_{\text{gr}}t^2\mathbb{1}$ 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 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, $\text{tr}(F(t))$ 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 3.3](#) we have

$$\begin{aligned} -t \frac{d}{dt} \log \det L(t) &= -t \cdot \text{tr} \left(L'(t) L(t)^{-1} \right) \\ &= -t \cdot \text{tr} \left((\mathbb{1}(\lambda_1 + \lambda_2) - A + 2\alpha_{\text{gr}} t \mathbb{1}) \cdot ((1 + \lambda_1 t)(1 + \lambda_2 t))^{-1} F(t) \right) \\ &= \frac{1}{(1 + \lambda_1 t)(1 + \lambda_2 t)} \text{tr} \left(-(\lambda_1 + \lambda_2) F(t) t + A F(t) t - 2\alpha_{\text{gr}} F(t) t^2 \right) \end{aligned}$$

where we used [Corollary 3.6](#). Now using [Lemma 3.5](#) again we may infer

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

combining the previous two identities yields

$$-t \frac{d}{dt} \log \det L(t) = \text{tr} \left(\frac{1 - \alpha_{\text{gr}} t^2}{(1 + \lambda_1 t)(1 + \lambda_2 t)} F(t) - \mathbb{1} \right). \quad (3)$$

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

$$\text{tr}(F(t)) \stackrel{?}{=} \sum_{k \geq 0} \text{tr}(B^k) t^k, \quad (4)$$

as $\text{tr}(B^k)$ is the weight of closed length- k circuits of direct edges in the nomadic world. However this is not quite right: $\text{tr}(B^k)$ 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., $\text{tr}(B^k)$ would fail to count the following:

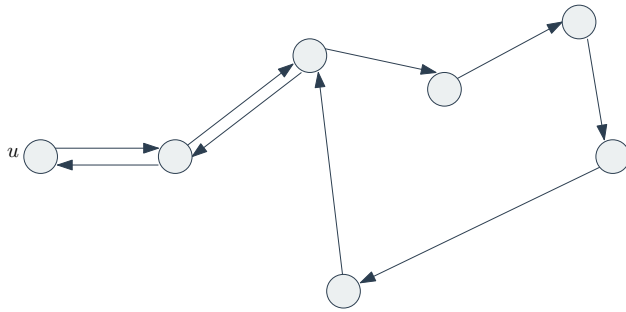


Figure 8: A length-9 nomadic walk from u to u with a *tail* of length 2

Thus we need to correct (4).

Definition 3.7. With the $-\mathbb{1}$ taking care of the omission of $k = 0$, we define

$$\text{Tails}(t) = \sum_{k \geq 1} (\text{weight of nomadic circuits of length } k) t^k = \text{tr}(F(t) - \mathbb{1}). \quad (5)$$

We also define

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

and

$$\begin{aligned} \text{Simple}(t) &= \sum_{k \geq 1} (\text{weight of non-stretched, tail-less nomadic circuits of length } k) t^k \\ &= \sum_{k \geq 1} \text{tr}(B^k) t^k = -t \frac{d}{dt} \log \det(\mathbb{1} - Bt), \quad (6) \end{aligned}$$

where the last equality used [Corollary 3.4](#).

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

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

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 2.11](#) gives

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

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

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

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

$$\begin{aligned}
\text{Tails}(t) &= \left(1 + (-\lambda_1\lambda_2)(c-2)t^2 + (-\lambda_1\lambda_2)^2(c-2)(c-1)t^4 + \dots\right)\text{Simple}(t) \\
&\quad + \left(1 + (-\lambda_1\lambda_2)(c-1)t^2 + (-\lambda_1\lambda_2)^2(c-1)^2t^4 + \dots\right)(\text{NoTails}(t) - \text{Simple}(t)) \\
&= \frac{1 - (-\lambda_1\lambda_2)t^2}{1 - (c-1)(-\lambda_1\lambda_2)t^2}\text{Simple}(t) + \frac{(\lambda_1 + \lambda_2)t}{1 - (c-1)(-\lambda_1\lambda_2)t^2}\text{Simple}(t) \\
&\iff \text{Simple}(t) = \frac{1 - \alpha_{\text{gr}}t^2}{(1 + \lambda_1t)(1 + \lambda_2t)}\text{Tails}(t). \quad (7)
\end{aligned}$$

Using $\text{Tails}(t) = \text{tr}(F(t) - \mathbb{1})$ (i.e., (5)), we obtain:

Corollary 3.8. $\text{Simple}(t) = \text{tr}\left(\frac{1 - \alpha_{\text{gr}}t^2}{(1 + \lambda_1t)(1 + \lambda_2t)}(F(t) - \mathbb{1})\right).$

But this is *almost* the same as (3). The difference is

$$\begin{aligned}
\text{tr}\left(\mathbb{1} - \frac{1 - \alpha_{\text{gr}}t^2}{(1 + \lambda_1t)(1 + \lambda_2t)}\mathbb{1}\right) &= \text{tr}\left(\frac{(\lambda_1 + \lambda_2)t + (c-2)(-\lambda_1\lambda_2)t^2}{(1 + \lambda_1t)(1 + \lambda_2t)}\mathbb{1}\right) \\
&= |V| \cdot \frac{(\lambda_1 + \lambda_2)t + (c-2)(-\lambda_1\lambda_2)t^2}{(1 + \lambda_1t)(1 + \lambda_2t)}.
\end{aligned}$$

Combining the above with (3), Corollary 3.8, and (6), we finally conclude

$$-t \frac{d}{dt} \log \det L(t) + |V| \cdot \frac{(\lambda_1 + \lambda_2)t + (c-2)(-\lambda_1\lambda_2)t^2}{(1 + \lambda_1t)(1 + \lambda_2t)} = -t \frac{d}{dt} \log \det(\mathbb{1} - Bt).$$

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

$$(\text{const.}) \cdot (1 + \lambda_1t)^{|V|\frac{c\lambda_2}{\lambda_2-\lambda_1}-1} (1 + \lambda_2t)^{|V|\frac{c\lambda_1}{\lambda_1-\lambda_2}-1} \det L(t) = \det(\mathbb{1} - Bt).$$

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

4 Connecting the adjacency and nomadic spectrum

Let $\mathcal{A} = (A_1, \dots, A_c)$ 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 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 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, \bar{x} such that $x\bar{x} = -1$.

Finally, the spectrum of B also contains 0 with multiplicity $2|E| - |V| \left(2 + \left(\frac{c\lambda_1}{\lambda_1 - \lambda_2} - 1 \right) + \left(\frac{c\lambda_2}{\lambda_2 - \lambda_1} - 1 \right) \right)$, 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 $|V| \left(2 + \left(\frac{c\lambda_1}{\lambda_1 - \lambda_2} - 1 \right) + \left(\frac{c\lambda_2}{\lambda_2 - \lambda_1} - 1 \right) \right)$ roots.

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

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

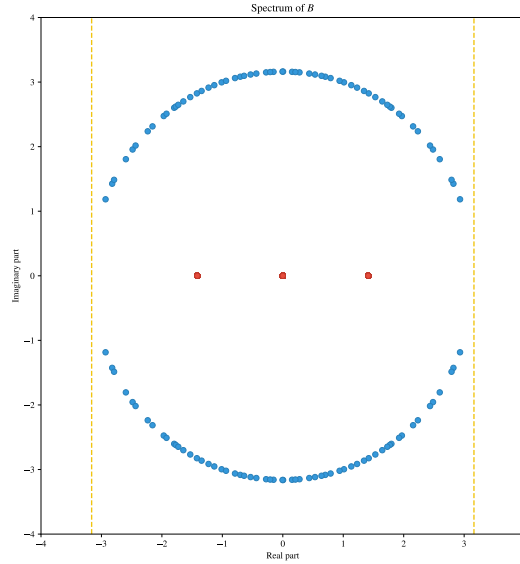


Figure 9: The spectrum of B for a additive 15-lift of 6 copies of a Sort_4 graph. The blue dots are eigenvalues that come from eigenvalues of A , the red dots are either $-\lambda_1$, $-\lambda_2$ or 0 and the yellow line is the limit $\sqrt{\alpha_{\text{gr}}}$.

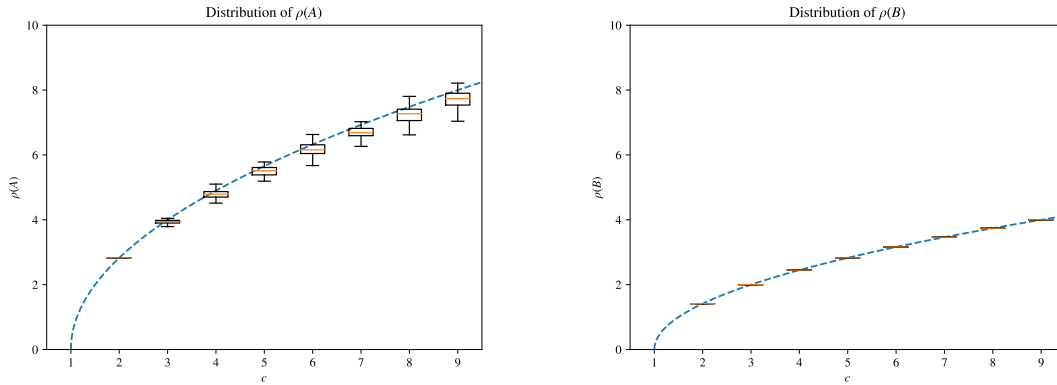


Figure 10: A box plot of $\rho(A)$ and $\rho(B)$ of 100 samples of random instance graphs as a function of c with $n = 15$, $r = 4$ and all atoms are the Sort_4 graph. The dashed line shows the theoretical bound prediction of $2\sqrt{\alpha_{\text{gr}}}$ for A and $\sqrt{\alpha_{\text{gr}}}$ for B .

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 9](#).

We can now prove the central theorem of this section:

Theorem 4.1. *Let \mathcal{I}_n be a random additive n -lift of \mathcal{A} with adjacency matrix $A_{\mathcal{I}_n}$, and let $\epsilon > 0$. Then:*

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

Proof. First recall [Theorem 1.6](#) (for fully formal statement, see [Theorem 6.20](#)) and notice that $\rho(|B|) = \alpha_{\text{gr}}$, which follows by using the trivial upper bound of α_{gr}^{2k} on $\text{tr}(|B|^k (|B|^*)^k)$. From cases 1 and 2 in the previous analysis we get that if $\rho(A_{\mathcal{I}_n}) \notin [\lambda_1 + \lambda_2 - 2\sqrt{\alpha_{\text{gr}}} - \varepsilon, \lambda_1 + \lambda_2 + 2\sqrt{\alpha_{\text{gr}}} + \varepsilon]$ there is some constant δ such that $\rho(B_n) > \sqrt{\alpha_{\text{gr}}} + \delta$, which happens with $o_{n \rightarrow \infty}(1)$ probability by [Theorem 6.20](#). \square

Also, we note that even though throughout our proof we hide various constant factors, the bounds obtained in [Theorem 4.1](#) and [Theorem 6.20](#) are empirically visible for very small values of n and c . To justify this claim we show in [Figure 10](#) a plot of samples of random instance graphs for different values of c with a fixed small n .

5 Additive products of 2-eigenvalue atoms

In this section, we let $\mathcal{A} = (A_1, \dots, A_c)$ 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 $X := A_1 \spadesuit \dots \spadesuit A_c$ be the additive product graph. We use A_X to denote the adjacency operator of X . In this section, \mathcal{I}_n is the instance graph of a random additive n -lift of \mathcal{A} with negations, and we use $A_{\mathcal{I}_n}$ to denote its adjacency matrix. Finally, we recall $\alpha_{\text{gr}} := (c-1)(-\lambda_1\lambda_2)$ and define the quantity $r_X := 2\sqrt{\alpha_{\text{gr}}}$.

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

Theorem 5.1. *The following are true about the spectrum of X :*

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

Theorem 5.2. *For every $\varepsilon > 0$, for large enough n , there are $|V(\mathcal{I}_n)| \times |V(\mathcal{I}_n)|$ positive semidefinite matrices M_+ and M_- with all-ones diagonals such that*

$$\begin{aligned} \langle A_{\mathcal{I}_n}, M_+ \rangle &\geq (\lambda_1 + \lambda_2 + r_X - \varepsilon)n \\ \langle A_{\mathcal{I}_n}, M_- \rangle &\leq (\lambda_1 + \lambda_2 - r_X + \varepsilon)n. \end{aligned}$$

with probability $1 - o_n(1)$.

In this section, when we measure the distance between vertices u and v in an instance graph \mathcal{I}_n , we look at the corresponding vertices in the constraint graph \mathcal{H} , and define $d(u, v) := \frac{d_{\mathcal{H}}(u, v)}{2}$. 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 5.3. *For all vertices v in $V(X)$, for $t \geq 1$ 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 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 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

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

Corollary 5.4. *Since all the weights of X are $\{\pm 1\}$ -valued, the degree of every vertex in X equals $c(-\lambda_1 \lambda_2)$.*

5.1 Enclosing the spectrum

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

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

The first part of the proof will involve showing that the spectral radius of B_X is bounded by $\sqrt{\alpha_{\text{gr}}}$, and the second part translates this bound to the desired one on $\text{Spec}(A_X)$. 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. $\text{Spec}(B_X) \subseteq [-\sqrt{\alpha_{\text{gr}}}, \sqrt{\alpha_{\text{gr}}}]$.

Proof. Arbitrarily fix a root r of X . Recall that the spectral radius of B_X is equal to $\lim (\|B_X^k\|_{\text{op}})^{1/k}$, and hence it suffices to bound $|\langle g, B_X^k f \rangle|$ 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 $e, e_1, \dots, e_{k-1}, e'$ is an (i, k) -nomadic walk, let $a(e, e') := \alpha_{\text{gr}}^{k/2-i}$. From [Lemma 5.3](#), the number of (i, k) -nomadic walks starting at a fixed e is at most $\frac{c}{c-1} \alpha_{\text{gr}}^{k-i}$. Similarly, the number of (i, k) -nomadic walks ending at fixed e' is at most $\frac{c}{c-1} \alpha_{\text{gr}}^i$. Now, we are ready to bound $|\langle g, B_X^k f \rangle|$ by imitating the proof of [[AFH15](#), Theorem 4.2].

$$\begin{aligned} |\langle g, B_X^k f \rangle| &\leq \left| \sum_{e, e_1, \dots, e_{k-1}, e' \text{ nomadic}} f(e') g(e) \right| \\ &\leq \sum_{e, e_1, \dots, e_{k-1}, e' \text{ nomadic}} |f(e') g(e)| \\ &\leq \sum_{e, e_1, \dots, e_{k-1}, e' \text{ nomadic}} a(e, e') f(e')^2 + \frac{1}{a(e, e')} g(e)^2 \\ &\leq \sup_{e'} \left(\sum_{e, e_1, \dots, e_{k-1}, e' \text{ nomadic}} a(e, e') \right) \|f\|_2^2 + \sup_e \left(\sum_{e, e_1, \dots, e_{k-1}, e' \text{ nomadic}} \frac{1}{a(e, e')} \right) \|g\|_2^2 \\ &\leq \sum_{i=0}^k \sup_{e'} \left(\sum_{(i, k)\text{-nomadic walks ending at } e'} a(e, e') \right) + \sup_e \left(\sum_{(i, k)\text{-nomadic walks starting at } e} \frac{1}{a(e, e')} \right) \end{aligned}$$

$$\begin{aligned}
&\leq \sum_{i=0}^k \alpha_{\text{gr}}^{k/2-i} \cdot \frac{c}{c-1} \alpha_{\text{gr}}^i + \sum_{i=0}^k \alpha_{\text{gr}}^{i-k/2} \cdot \frac{c}{c-1} \alpha_{\text{gr}}^{k-i} \\
&= \frac{2kc}{c-1} \alpha_{\text{gr}}^{k/2}
\end{aligned}$$

Thus, we have

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

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

Lemma 5.6. *If 0 is an approximate eigenvalue of $Q_t := (t^2 + (c-1)(-\lambda_1\lambda_2))\mathbb{1} - A_X t + (\lambda_1 + \lambda_2)\mathbb{1}t$, 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,

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

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

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

Using [Fact 2.10](#) and [Fact 2.11](#), the first term of the three above can be rewritten as

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

which lets us continue the chain of equalities

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

$$\begin{aligned}
&= (c-1)(-\lambda_1\lambda_2)f_v - t(Af)_v + t(\lambda_1 + \lambda_2 + t)f_v \\
&= (Q_t f)_v.
\end{aligned}$$

Thus,

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

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

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

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

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

Proof of Item 1 in Theorem 5.1. Let Q_t be as defined in the statement of Lemma 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 5.4 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 5.6 that

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

is an approximate eigenvalue of B_X . Since $\text{Spec}(B_X)$ is contained in $[-\sqrt{\alpha_{\text{gr}}}, \sqrt{\text{gr}}]$, 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 $\text{Spec}(A_X)$ 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 $\text{Spec}(A_X)$. 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 $\text{Spec}(A_X)$, it is an isolated point in the spectrum, and hence, by Theorem 2.25, is an eigenvalue of A_X , which means d_X cannot be in $\text{Spec}(A_X)$. \square

5.2 Construction of Witness Vectors

Lemma 5.7 (Item 2 of Theorem 5.1 restated). *There exists $\lambda_- \leq \lambda_1 + \lambda_2 - r_X$ and $\lambda_+ \geq \lambda_1 + \lambda_2 + r_X$ 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.

$$f_v^{(s)}(u) := \rho(s)^{d(u,v)} \prod_{\{i,j\} \in \mathcal{P}_{uv}} (A_X)_{ij} \quad \text{where } \mathcal{P}_{uv} \text{ is the unique nomadic walk between } u \text{ and } v \quad (8)$$

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 $\langle f_v^{(s)}, A_X f_v^{(s)} \rangle$.

$$\begin{aligned} \langle f_v^{(s)}, A_X f_v^{(s)} \rangle &= \sum_{u \in V} f_v^{(s)}(u) A_X f_v^{(s)}(u) \\ &= f_v^{(s)}(v) \sum_{w \in N(v)} (A_X)_{vw} f_v^{(s)}(w) + \sum_{u \in V, u \neq v} f_v^{(s)}(u) \sum_{w \in N(u)} (A_X)_{uw} f_v^{(s)}(w) \\ &= \sum_{w \in N(v)} (A_X)_{vw}^2 \rho(s) + \sum_{u \in V, u \neq v} f_v^{(s)}(u) \sum_{w \in N(u)} (A_X)_{uw} f_v^{(s)}(w) \end{aligned} \quad (9)$$

Let $w_0, w_1, \dots, w_{T-1}, w_T$ 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 2.10](#) on the first term of the above, we get

$$\begin{aligned} (9) &= c(-\lambda_1\lambda_2)\rho(s) + \sum_{u \in V, u \neq v} \rho(s) W_{u^*v} (A_X)_{uu^*} \cdot \\ &\quad \left((A_X)_{uu^*} W_{u^*v} + \sum_{w \in \text{Atom}(\{u^*, u\})} \rho(s) (A_X)_{u^*w} (A_X)_{wu} W_{u^*v} + \sum_{\substack{w \notin \text{Atom}(\{u, u^*\}) \\ w \in N(u)}} \rho(s)^2 (A_X)_{u^*u} (A_X)_{uw}^2 W_{u^*v} \right) \\ &= c(-\lambda_1\lambda_2)\rho(s) + \sum_{u \in V, u \neq v} \rho(s) W_{u^*v}^2 (A_X)_{uu^*}^2 \cdot \\ &\quad \left(1 + \frac{\sum_{w \in \text{Atom}(\{u^*, u\})} \rho(s) (A_X)_{u^*w} (A_X)_{wu}}{A_{uu^*}} + \sum_{\substack{w \notin \text{Atom}(\{u, u^*\}) \\ w \in N(u)}} (A_X)_{uw}^2 \rho(s)^2 \right) \end{aligned}$$

Now we apply [Fact 2.10](#) and [Fact 2.11](#) and get

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

When $s = \pm 1$, the above quantity is equal to

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

Now, note that

$$\frac{\langle f_v^{(s)}, A_X f_v^{(s)} \rangle}{\langle f_v^{(s)}, f_v^{(s)} \rangle} = \frac{c(-\lambda_1\lambda_2)\rho(s)}{\|f_v^{(s)}\|^2} + \left(1 - \frac{1}{\|f_v^{(s)}\|^2} \right) \cdot \left(\frac{1 + (1-\delta)^2}{\rho(s)} + (\lambda_1 + \lambda_2) \right) \quad (10)$$

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

$$\begin{aligned}
\|f_v^{(s)}\|^2 &= \sum_{t=0}^{\infty} \rho(s)^{2t} \sum_{u:d(u,v)=t} \prod_{\{i,j\} \in \mathcal{P}_{uv}} (A_X)_{ij}^2 \\
&= \sum_{t=0}^{\infty} \rho(s)^{2t} c(c-1)^{t-1} (-\lambda_1\lambda_2)^t && \text{(by [Lemma 5.3](#))} \\
&= \frac{c}{c-1} \sum_{t=0}^{\infty} \left(\frac{(1-\delta)^{2t}}{(c-1)^t (-\lambda_1\lambda_2)^t} \right) (c-1)^t (-\lambda_1\lambda_2)^t \\
&= \frac{c}{c-1} \sum_{t=0}^{\infty} (1-\delta)^{2t} \\
&= \frac{c}{c-1} \cdot \frac{1}{\delta(2-\delta)}
\end{aligned}$$

Plugging this back in to [\(10\)](#) gives

$$\begin{aligned}
(10) &= \delta(2-\delta)(c-1)(-\lambda_1\lambda_2)\rho(s) + \left(\frac{1 + (1-\delta)^2}{\rho(s)} + (\lambda_1 + \lambda_2) \right) \cdot \left(1 - \frac{(c-1)\delta(2-\delta)}{c} \right) \\
&= \delta(2-\delta)s(1-\delta)\sqrt{(c-1)(-\lambda_1\lambda_2)} + \\
&\quad \left((1 + (1-\delta)^2)\sqrt{(c-1)(-\lambda_1\lambda_2)} \frac{1}{s(1-\delta)} + (\lambda_1 + \lambda_2) \right) \cdot \left(1 - \frac{(c-1)\delta(2-\delta)}{c} \right)
\end{aligned}$$

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 $s = -1$.

□

5.3 SDP solution for random additive lifts

For $\varepsilon > 0$, consider $f_v^{(1)}$ constructed in the proof of [Lemma 5.7](#), for which

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

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

$$\langle g_v, A_X g_v \rangle \geq \lambda_1 + \lambda_2 + r_X - \varepsilon' \quad (11)$$

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

$$g_v(u) = \begin{cases} 0 & \text{if } d(u, v) > L_{\varepsilon'} \\ C \prod_{\{i,j\} \in \mathcal{P}_{uv}} \frac{(1-\delta)(A_X)_{ij}}{\sqrt{(c-1)(-\lambda_1\lambda_2)}} & \mathcal{P}_{uv} \text{ unique shortest nomadic walk from } u \text{ to } v \end{cases}$$

where C is a constant chosen so that g_v has unit norm.

Recall that \mathcal{I}_n is a random signed additive n -lift obtained from a sequence of atoms \mathcal{A} .

Definition 5.8. Let G be a graph and let $\phi : E(G) \rightarrow \{\pm 1\}$ be a signing of the edges. We call a signing ϕ *balanced* if for any cycle given by sequence of edges e_1, \dots, 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 5.9. *Suppose ϕ is a balanced signing of G . Then there exists a diagonal sign operator D such that $A_{\phi(G)} = DA_G D^\dagger$.*

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 $DA_G D^\dagger = A_{\phi(G)}$. 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,

$$(A_{\phi(G)})_{ij} = \phi(\{i, j\})(A_G)_{ij} = D_{ii} D_{jj} (A_G)_{ij} = (DA_G D^\dagger)_{ij}$$

which proves the claim. \square

Lemma 5.10. *Let X_D be the graph with the adjacency operator $DA_X D^\dagger$ where D is a diagonal sign matrix. There exists D such that X_D covers \mathcal{I}_n .*

Proof. When \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 $\tilde{\mathcal{I}}_n$ between (i) and (ii) is covered by X via a map $\pi : V(X) \rightarrow V(\tilde{\mathcal{I}}_n)$. 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 $e \in E(\mathcal{I}_n)$. X' can be seen as a balanced signing applied on X , and hence there exists such a D by [Lemma 5.9](#). \square

Definition 5.11. Let π be a covering map from appropriate X_D to \mathcal{I}_n . Call a vertex $v \in V(\mathcal{I}_n)$ 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 5.12. The condition of a vertex v in $V(\mathcal{I}_n)$ being L -bad according to [Definition 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 5.12](#) in hand, we can extract the following as a consequence of [\[DMO⁺19\]](#).

Lemma 5.13. *The number of K -bad vertices in graph \mathcal{I}_n for constant K is bounded by $O(\log n)$ with probability $1 - o_n(1)$.*

Construct a vector \tilde{g}_v for each vertex v of \mathcal{I}_n .

$$\tilde{g}_v = \begin{cases} e_v & \text{if } v \text{ is } L_{\epsilon'}\text{-bad} \\ g_v & \text{otherwise} \end{cases}$$

We are finally ready to prove [Theorem 5.2](#).

Proof of Theorem 5.2. Let

$$M_+ := \sum_{v \in V(\mathcal{I}_n)} \tilde{g}_v \tilde{g}_v^+$$

Writing out $(M_+)_{uu}$ for arbitrary u

$$\begin{aligned} (M_+)_{uu} &= \sum_{v \in V(\mathcal{I}_n)} \tilde{g}_v(u) \tilde{g}_v(u) \\ &= \sum_{v \in V(\mathcal{I}_n)} \tilde{g}_u(v)^2 \\ &= \|\tilde{g}_u\|^2 = 1 \end{aligned}$$

and writing out $\langle A_{\mathcal{I}_n}, M_+ \rangle$ gives the following with probability $1 - o_n(1)$.

$$\begin{aligned} \langle A_{\mathcal{I}_n}, M_+ \rangle &= \sum_{v \in V(\mathcal{I}_n)} \langle \tilde{g}_v, A_{\mathcal{I}_n} \tilde{g}_v \rangle \\ &= \sum_{\substack{v \in V(\mathcal{I}_n) \\ v \text{ is not } (L_\epsilon + 1)\text{-bad}}} \langle \tilde{g}_v, A_{\mathcal{I}_n} \tilde{g}_v \rangle + \sum_{\substack{v \in V(\mathcal{I}_n) \\ v \text{ is } (L_\epsilon + 1)\text{-bad}}} \langle \tilde{g}_v, A_{\mathcal{I}_n} \tilde{g}_v \rangle \\ &\geq \sum_{\substack{v \in V(\mathcal{I}_n) \\ v \text{ is not } (L_\epsilon + 1)\text{-bad}}} \lambda_1 + \lambda_2 + r_X - \epsilon' + \sum_{\substack{v \in V(\mathcal{I}_n) \\ v \text{ is } (L_\epsilon + 1)\text{-bad}}} c(\lambda_1 \lambda_2) && \text{(by (11))} \\ &\geq (n - O(\log n))(\lambda_1 + \lambda_2 + r_X - \epsilon') - O(\log n) && \text{(by Lemma 5.13)} \end{aligned}$$

$$= (1 - o_n(1))(\lambda_1 + \lambda_2 + r_X - \varepsilon')n$$

The desired inequality on $\langle A_{\mathcal{I}_n}, M_+ \rangle$ can be obtained by choosing ε' small enough and n large enough. The inequality on $\langle A_{\mathcal{I}_n}, M_- \rangle$ can be proved by repeating the whole section and proof by constructing vectors \tilde{g}_v from $f_v^{(-1)}$. \square

6 Friedman/Bordenave for additive lifts

Theorem 6.1. *Let $\mathcal{A} = (A_1, \dots, A_c)$ 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 \mathcal{H}_n denote a random n -lifted constraint graph and $\mathcal{I}_n = \mathcal{A}(\mathcal{H}_n)$ an associated instance graph with 1-wise uniform negations $(\zeta_{ii'}^f)$. Finally, let B_n denote the nomadic walk matrix for \mathcal{I}_n . Then for every constant $\varepsilon > 0$,*

$$\Pr[\rho(B_n) \geq \sqrt{\rho(|B_1|)} + \varepsilon] \leq \delta,$$

where $\delta = \delta(n)$ is $o_{n \rightarrow \infty}(1)$.

Remark 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 $\rho(B_n)$ 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 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 $\zeta_{ii'}^f$ 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 6.1](#) involves applying the Trace Method to B_n . Since B_n is not a self-adjoint operator, a natural way to do this is to consider $\text{tr}(B_n^\ell B_n^{*\ell})$ for some large ℓ . Roughly speaking, this counts the number of closed walks that walk nomadically in \mathcal{I}_n for the first ℓ steps, and then walk nomadically in the *reverse* of \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 $\text{tr}((B_n^\ell B_n^{*\ell})^k)$. 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⁺19] and [BDH18]. However, our main technical lemma, Lemma 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.

6.1 Trace Method setup, and getting rid of tangles

To begin carrying out this proof strategy, we first define tangle-freeness.

Definition 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 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 \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:

$$\begin{aligned} \text{tr}((\mathbf{B}_n^\ell \mathbf{B}_n^{*\ell})^k) &= \sum_{\vec{e}_0, \dots, \vec{e}_{2k\ell-1}, \vec{e}_{2k\ell} = \vec{e}_0} \mathbf{B}_n[\vec{e}_0, \vec{e}_1] \cdots \mathbf{B}_n[\vec{e}_{\ell-1}, \vec{e}_\ell] \mathbf{B}_n^*[\vec{e}_\ell, \vec{e}_{\ell+1}] \cdots \mathbf{B}_n^*[\vec{e}_{2\ell-1}, \vec{e}_{2\ell}] \cdots \mathbf{B}_n^*[\vec{e}_{2k\ell-1}, \vec{e}_{2k\ell}] \\ &= \sum_{\vec{e}_0, \dots, \vec{e}_{2k\ell-1}, \vec{e}_{2k\ell} = \vec{e}_0} \mathbf{B}_n[\vec{e}_0, \vec{e}_1] \cdots \mathbf{B}_n[\vec{e}_{\ell-1}, \vec{e}_\ell] \mathbf{B}_n[\vec{e}_{\ell+1}, \vec{e}_\ell] \cdots \mathbf{B}_n[\vec{e}_{2\ell}, \vec{e}_{2\ell-1}] \cdots \mathbf{B}_n[\vec{e}_{2k\ell}, \vec{e}_{2k\ell-1}] \\ &= \sum \text{wt}(e_1) N_{\vec{e}_0, \vec{e}_1} \cdots \text{wt}(e_\ell) N_{\vec{e}_{\ell-1}, \vec{e}_\ell} \text{wt}(e_\ell) N_{\vec{e}_\ell, \vec{e}_{\ell+1}} \cdots \text{wt}(e_{2\ell-1}) N_{\vec{e}_{2\ell-1}, \vec{e}_{2\ell}} \cdots \text{wt}(e_{2k\ell-1}) N_{\vec{e}_{2k\ell-1}, \vec{e}_{2k\ell}} \end{aligned} \quad (12)$$

where $\text{wt}(e)$ 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 \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 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

¹¹We chose the factor 4 here for “safety”. For quantitative aspects of our theorem, constant factors on ℓ will be essentially costless.

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 $\text{wt}(\mathcal{L})$, is the product of the signs of the non-spur edges in \mathcal{L} .

Definition 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 $\text{wt}(\mathcal{C}) \in \{\pm 1\}$ for the weight of the associated nomadic linkage in the instance graph.

Given these definitions, (12) tells us:

$$\text{tr}((\mathbf{B}_n^\ell \mathbf{B}_n^{*\ell})^k) = \sum_{\substack{(2k \times 2\ell)\text{-nonbacktracking} \\ \mathcal{A}\text{-linkages } \mathcal{C} \text{ in } \mathcal{H}_n}} \text{wt}(\mathcal{C}). \quad (13)$$

Next, we make the observation that if \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 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 6.5](#) we have:

Lemma 6.9. *Provided $\ell \leq \kappa \log n$ for a certain universal $\kappa > 0$, we get that $\text{tr}((\mathbf{B}_n^\ell \mathbf{B}_n^{*\ell})^k) = S$ holds except with probability $O(1/n^{99})$, where*

$$S := \sum_{\substack{(2k \times 2\ell)\text{-nonbacktracking} \\ \text{externally } \ell\text{-tangle-free} \\ \mathcal{A}\text{-linkages } \mathcal{C} \text{ in } \mathcal{H}_n}} \text{wt}(\mathcal{C}).$$

In order to apply Markov’s inequality later, we will need the following technical claim:

Claim 6.10. S is a nonnegative random variable.

Proof. Given \mathcal{I}_n , recall that

$$\mathbf{B}_n^\ell[\vec{e}, \vec{f}] = \sum_{\substack{\text{nomadic walks} \\ \vec{e}=\vec{e}_0, \vec{e}_1, \dots, \vec{e}_\ell=\vec{f} \text{ in } \mathcal{I}_n}} \text{wt}(e_1)\text{wt}(e_2) \cdots \text{wt}(e_\ell).$$

Using a key idea of Bordenave (based on the “selective trace” of Friedman), define the related operator $\mathbf{B}_n^{(\ell)}$ via

$$\mathbf{B}_n^{(\ell)}[\vec{e}, \vec{f}] = \sum_{\substack{\text{externally } \ell\text{-tangle-free nomadic walks} \\ \vec{e}=\vec{e}_0, \vec{e}_1, \dots, \vec{e}_\ell=\vec{f} \text{ in } \mathcal{I}_n}} \text{wt}(e_1)\text{wt}(e_2) \cdots \text{wt}(e_\ell),$$

where again the walk is said to be “externally ℓ -tangle-free” if every vertex it touches is ℓ -tangle-free with \mathcal{H}_n . Then very similar to the analysis that gave us (12) and (13), we get that

$$\mathbf{S} = \text{tr}((\mathbf{B}_n^{(\ell)}(\mathbf{B}_n^{(\ell)})^*)^k).$$

Thus \mathbf{S} is visibly always nonnegative, being the trace of the k th power of the positive semidefinite matrix $\mathbf{B}_n^{(\ell)}(\mathbf{B}_n^{(\ell)})^*$. \square

With these results in place, we can proceed to the main goal of the Trace Method: bounding $\mathbf{E}[\mathbf{S}]$. Such a bound can be used in the following lemma:

Lemma 6.11. *Assume that $\ell \leq \kappa \log n$ and $k\ell = \omega(\log n)$. Then from $\mathbf{E}[\mathbf{S}] \leq R$ we may conclude that $\rho(\mathbf{B}_n) \leq (1 + o_n(1)) \cdot R^{\frac{1}{2k\ell}}$ holds, except with probability $O(1/n^{99})$.*

Proof. Let $\mathbf{T} = \text{tr}((\mathbf{B}_n^\ell \mathbf{B}_n^{*\ell})^k)$. On one hand, with λ denoting eigenvalues and σ denoting singular values, we have

$$\mathbf{T} \geq \lambda_{\max}((\mathbf{B}_n^\ell \mathbf{B}_n^{*\ell})^k) = \lambda_{\max}\left(\sqrt{\mathbf{B}_n^\ell \mathbf{B}_n^{*\ell}}\right)^{2k} = \sigma_{\max}(\mathbf{B}_n^\ell)^{2k} \geq \rho(\mathbf{B}_n^\ell)^{2k} = \rho(\mathbf{B}_n)^{2k\ell}.$$

On the other hand, since \mathbf{S} is a nonnegative random variable (Claim 6.10), we can apply Markov’s Inequality to deduce that $\mathbf{S} \leq n \cdot R$ except with probability at most $1/n$. Now from Lemma 6.9 we may infer that except with probability $O(1/n^{99})$,

$$\mathbf{T} = \mathbf{S} \leq n \cdot R \quad \implies \quad \rho(\mathbf{B}_n)^{2k\ell} \leq n \cdot R.$$

The result now follows by taking $2k\ell$ -th roots. \square

6.2 Eliminating singletons, and reduction to counting

Our next step toward bounding $\mathbf{E}[\mathbf{S}]$ is typical of the Trace Method: Rather than first choosing \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

$\mathcal{K}_n =$ the “complete” constraint graph with cn constraint vertices and rn variable vertices,

this means that

$$S = \sum_{\substack{(2k \times 2\ell)\text{-nonbacktracking} \\ \mathcal{A}\text{-linkages } \mathcal{C} \text{ in } \mathcal{K}_n}} 1[\mathcal{C} \text{ is in } \mathcal{H}_n] \cdot 1[\mathcal{C} \text{ is externally } \ell\text{-tangle-free within } \mathcal{H}_n] \cdot \text{wt}_{\mathcal{I}_n}(\mathcal{C}). \quad (14)$$

Here we wrote $\text{wt}_{\mathcal{I}_n}(\mathcal{C})$ to emphasize that even once \mathcal{C} is in \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 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 (14), consider $\mathbf{E}[S]$. If \mathcal{C} contains any singleton, then it will contribute 0 to this expectation. The reason is that, provided \mathcal{C} appears in \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 $\text{wt}_{\mathcal{I}_n}(\mathcal{C})$. On the other hand, when \mathcal{C} is singleton-free, we will simply upper-bound the (conditional) expectation of $\text{wt}_{\mathcal{I}_n}(\mathcal{C})$ by $+1$. We conclude that

$$\mathbf{E}[S] \leq \sum_{\substack{(2k \times 2\ell)\text{-nonbacktracking} \\ \text{singleton-free} \\ \mathcal{A}\text{-linkages } \mathcal{C} \text{ in } \mathcal{K}_n}} \Pr[\mathcal{C} \text{ is in } \mathcal{H}_n \text{ and is externally } \ell\text{-tangle-free therein}]. \quad (15)$$

Let us now begin to simplify the probability calculation.

Definition 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 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:

$$\begin{aligned} \text{linkage } \mathcal{C} \text{ not even internally } \ell\text{-tangle-free} \\ \implies \Pr[\mathcal{C} \text{ is in } \mathcal{H}_n \text{ and is externally } \ell\text{-tangle-free therein}] = 0. \end{aligned}$$

Thus we can restrict the sum in (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

$$\mathbf{E}[S] \leq \sum_{\substack{(2k \times 2\ell)\text{-nonbacktracking} \\ \text{interally } \ell\text{-tangle-free, singleton-free} \\ \mathcal{A}\text{-linkages } \mathcal{C} \text{ in } \mathcal{K}_n}} \Pr[\mathcal{C} \text{ is in } \mathcal{H}_n]. \quad (16)$$

We will now bound $\Pr[\mathcal{C} \text{ is in } \mathcal{H}_n]$, so as to reduce all our remaining problems to counting. Towards this, recall that \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 \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 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, $\Pr[\mathcal{C} \text{ is in } \mathcal{H}_n] = 0$ if \mathcal{C} is invalid. Thus from (16) we can deduce:

$$\mathbf{E}[\mathcal{S}] \leq \sum_{\substack{(2k \times 2\ell)\text{-nonbacktracking} \\ \text{valid, internally } \ell\text{-tangle-free, singleton-free} \\ \mathcal{A}\text{-linkages } \mathcal{C} \text{ in } \mathcal{K}_n}} \Pr[\mathcal{C} \text{ is in } \mathcal{H}_n]. \quad (17)$$

Next, it is straightforward to show the following lemma (see Proposition A.8 of [DMO⁺19] for essentially the same observation):

Lemma 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*

$$\Pr[\mathcal{C} \text{ is in } \mathcal{H}_n] = (1 + o_n(1)) \cdot n^{-|E(\mathcal{C})|}.$$

Proof. (Sketch.) Proceed through the edges in $E(\mathcal{C})$ in an arbitrary order. Each has approximately a $1/n$ chance of appearing in \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 \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 $|E(\mathcal{C})| \leq 4k\ell$, 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 $(1 + O(k\ell)/n)^{4k\ell} \leq 1 + o_n(1)$, the inequality using $(k\ell)^2 = o(n)$. \square

Combining this lemma with (17) and Lemma 6.11, we are able to reduce bounding $\rho(\mathbf{B}_n)$ to a counting problem:

Lemma 6.17. *Assume that $\ell \leq \kappa \log n$ and $\omega(\log n) < k\ell < o(\sqrt{n})$. Then except with probability $O(1/n^{99})$,*

$$\rho(\mathbf{B}_n) \leq (1 + o_n(1)) \cdot R^{\frac{1}{2k\ell}}, \quad \text{where } R := \sum_{\substack{(2k \times 2\ell)\text{-nonbacktracking} \\ \text{valid, internally } \ell\text{-tangle-free, singleton-free} \\ \mathcal{A}\text{-linkages } \mathcal{C} \text{ in } \mathcal{K}_n}} n^{-|E(\mathcal{C})|}.$$

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 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 6.1](#). We begin with a graph-theoretic lemma and then deduce the nearly-duplicative property.

Lemma 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 $w \leq 4k\ell$. 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 \tilde{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 \tilde{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

$$\begin{aligned} 2 + O(\log(w)/\ell) + \frac{2t}{w} &\geq \frac{w_1 + 2w_2 + 3w_{3+}}{w} = \frac{w_1 + 2(w - w_1 - w_{3+}) + 3w_{3+}}{w} = 2 - \frac{w_1}{w} + \frac{w_{3+}}{w} \\ &\implies w_{3+} \leq O(w \log(w)/\ell) + w_1 + 2t. \end{aligned}$$

The first term here is $O(k \log(k\ell))$ as desired, since $w \leq 4k\ell$. 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 $w_1 \leq 2k$. Finally, $2t \leq O(k)$ follows from the below claim combined with $w \leq 4k\ell$:

$$t \leq \frac{w}{2\ell} + 1. \tag{18}$$

We establish [\(18\)](#) 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 $3\ell - 1 \geq 2\ell$ vertices from these paths. It follows that $w = |V| \geq 2\ell(t - 1)$, establishing [\(18\)](#). \square

Corollary 6.19. *In the setting of [Lemma 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 6.18](#) there are at most $O(k \log(k\ell))$ such constraint-vertices. Further, by validity each constraint-vertex can support at most $\binom{\ell}{2} = O(1)$ 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. \square

6.4 The final countdown

We now wish to count the objects summed in the definition of R from [Lemma 6.17](#). The remainder of this section will be devoted to proving:

Theorem 6.20. *For every $\varepsilon > 0$, except with probability $O(1/n^{99})$,*

$$\rho(\mathbf{B}_n) \leq (1 + o_n(1)) \cdot (1 + \varepsilon) \cdot \sqrt{\rho(|B_1|)}.$$

The bulk of the technical matter in the proof of [Theorem 6.20](#) will involve analyzing

$$(2k \times 2\ell)\text{-nonbacktracking, valid, internally } \ell\text{-tangle-free, singleton-free, } \mathcal{A}\text{-linkages } \mathcal{C} \quad (19)$$

in \mathcal{K}_n .

Definition 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 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 6.23. We write $\text{Lkgs}(f, b)$ for the collection of linkages as in [\(19\)](#) having exactly f fresh edges and b boundary edges.

Our goal is to show:

Lemma 6.24. *For every $\hat{\rho} > \rho(|B_1|)$ we have:*

$$|\text{Lkgs}(f, b)| \leq \text{poly}(k, \ell)^{b+k} \cdot n^{f+1} \cdot \hat{\rho}^{f/2}$$

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 6.24](#).

Given a linkage \mathcal{C} we write $\mathcal{C} = ((v_1, i_1), (v_2, i_2), \dots, (v_{4k(\ell+1)}, i_{4k(\ell+1)}))$, 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 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 $\mathcal{C} = ((v_1, i_1), \dots, (v_{4k(\ell+1)}, i_{4k(\ell+1)}))$ and $\mathcal{C}' = ((v'_1, i'_1), \dots, (v'_{4k(\ell+1)}, i'_{4k(\ell+1)}))$, 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 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 $(v, 2), \dots$, and finally (v, j) . We write $\text{Lkgs}^c(f, b)$ for the collection of *canonical* linkages as in [\(19\)](#) having exactly f fresh steps and b boundary steps.

Proposition 6.27. $|\text{Lkgs}(f, b)| \leq n^{f+1} |\text{Lkgs}^c(f, b)|$.

Proof. It suffices to show that for every canonical linkage $\mathcal{C} \in \text{Lkgs}^c(f, b)$, it has at most n^{f+1} isomomorphic linkages $\mathcal{C}' \in \text{Lkgs}(f, b)$. By [Fact 6.22](#), \mathcal{C} visits exactly $f + 1$ distinct vertices, call them $\{(v^{(1)}, i^{(1)}), \dots, (v^{(f+1)}, i^{(f+1)})\}$. Every isomorphic \mathcal{C}' may be obtained by taking a list of numbers $(i'_1, \dots, i'_{f+1}) \in [n]^{f+1}$ 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. \square

We now have all the tools to prove the desired lemma.

Proof of [Lemma 6.24](#). With [Proposition 6.27](#) in place, it suffices to bound the number of canonical linkages as follows:

$$|\text{Lkgs}^c(f, b)| \leq \text{poly}(k, \ell)^{b+k} \cdot \hat{\rho}^{f/2}.$$

Our strategy is to give an encoding of linkages in $\text{Lkgs}^c(f, b)$, and then bound the number of possible encodings. Let \mathcal{C} be an arbitrary linkage in $\text{Lkgs}^c(f, b)$. 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 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 $K_{r,\mathcal{C}}$.

Encoding boundary steps. Let β be a block of boundary steps $(v_0, v_1), \dots, (v_{|\beta|-1}, v_{|\beta|})$. 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 $(t_0, t_1), \dots, (t_{|\beta|-1}, t_{|\beta|})$. Let β_1, \dots, β_T be the blocks of boundary steps in the order in which they appear in \mathcal{C} . We store the concatenation of $\text{Enc}_b(\beta_1), \dots, \text{Enc}_b(\beta_T)$, 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 β_1, \dots, β_T be the blocks of stale steps in the order they appear in \mathcal{C} . We store the concatenation of $\text{Enc}_s(\beta_1), \dots, \text{Enc}_s(\beta_T)$, 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_{fresh} , P_{boundary} and P_{stale} , we can infer a partition of $[4k(\ell + 1)]$ into blocks in order β_1, \dots, β_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 β_1, \dots, β_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, \dots, 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 \tilde{W}_β be the maximal subwalk of W_β that starts and ends with a variable vertex. Note that \tilde{W}_β corresponds exactly to a nomadic walk in \mathcal{I}_1 whose length is at most $\lfloor |\beta|/2 \rfloor$. Now regarding W_β , either W_β is equal to \tilde{W}_β (there is 1 way in which this can happen), or both the first and last steps of W_β are not in \tilde{W}_β (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 \tilde{W}_β (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 $\delta_s \leq \|(|B_1|)^s\|_1$, where $\|(|B_1|)^s\|_1 = \sup\{\|(|B_1|)^s x\|_1 : \|x\|_1 = 1\}$.

To bound the above, first observe that we have a simple bound $\|(|B_1|)^s\|_1 \leq \kappa^s$ 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

$$\lim_{s \rightarrow \infty} (\|(|B_1|)^s\|_1)^{1/s} = \rho(|B_1|),$$

and hence for any $\hat{\rho} > \rho(|B_1|)$, there is a constant ℓ_0 such that $\|(|B_1|)^s\|_1 \leq (\hat{\rho})^s$ for all $s \geq \ell_0$. Putting these two bounds together we get that for any $s \geq \ell_0$,

$$\delta_s \leq \|(|B_1|)^s\|_1 \leq (\hat{\rho})^{s-\ell_0} \kappa^{\ell_0}.$$

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 $((c+1)^2 \kappa^{\ell_0} \hat{\rho}^{-\ell_0})^{b+2k} (\hat{\rho})^{f/2}$.

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 $M \leq 2$, 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 $r \geq 0$. 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:

$$(4k(\ell + 1))^{6b+8k} ((c + 1)^2 \kappa^{\ell_0} (\hat{\rho})^{-\ell_0})^{b+2k} (\hat{\rho})^{f/2} 2^{b+2k} (f + 1)^{3b+2k} \quad (20)$$

which, when combined with [Proposition 6.27](#) gives the desired claim. \square

We wrap everything up by combining the results of [Lemma 6.24](#) with [Lemma 6.17](#) to prove [Theorem 6.20](#).

Proof of Theorem 6.20. Let $\ell = \kappa \log n$, where κ is the universal constant from [Proposition 6.5](#), let k be chosen so that $k\ell = \omega(\log n)$, let R be as in [Lemma 6.17](#), and let $\hat{\rho}$ be any constant greater than $\rho(|B_1|)$. Then we have

$$\begin{aligned} R &= \sum_{\substack{(2k \times 2\ell)\text{-nonbacktracking} \\ \text{valid, internally } \ell\text{-tangle-free, singleton-free} \\ \mathcal{A}\text{-linkages } \mathcal{C} \text{ in } \mathcal{K}_n}} n^{-|E(\mathcal{C})|} \\ &= \sum_{f=0}^{\infty} \sum_{b=0}^{\infty} |\text{Lkgs}(f, b)| n^{-(f+b)} \\ &= \sum_{f=0}^{2k\ell + O(k \log(k\ell))} \sum_{b=0}^{\infty} |\text{Lkgs}(f, b)| n^{-(f+b)} && \text{(by Corollary 6.19)} \\ &\leq \sum_{f=0}^{2k\ell + O(k \log(k\ell))} \sum_{b=0}^{\infty} \frac{\text{poly}(k, \ell)^b \cdot \text{poly}(k, \ell)^k \cdot (\hat{\rho})^{f/2} \cdot n}{n^b} && \text{(by Lemma 6.24)} \\ &= \sum_{f=0}^{2k\ell + O(k \log(k\ell))} n \cdot \text{poly}(k, \ell)^k \cdot (\hat{\rho})^{f/2} \sum_{b=0}^{\infty} \left(\frac{\text{poly}(k, \ell)}{n} \right)^b \\ &= \sum_{f=0}^{2k\ell + O(k \log(k\ell))} n \cdot \text{poly}(k, \ell)^k \cdot (\hat{\rho})^{f/2} \cdot \left(\frac{1}{1 - \frac{\text{poly}(k, \ell)}{n}} \right) \\ &\leq 2n \cdot \text{poly}(k, \ell)^k (2k\ell + O(k \log(k\ell))) (\hat{\rho})^{k\ell + O(k \log(k\ell))} \end{aligned}$$

For the choice of k and ℓ in the theorem statement, we can use [Lemma 6.17](#) to conclude that

$$\rho(\mathbf{B}_n) \leq (1 + o_n(1)) \cdot \sqrt{\hat{\rho}}.$$

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

$$\rho(\mathbf{B}_n) \leq (1 + o_n(1)) \cdot (1 + \varepsilon) \cdot \sqrt{\rho(|B_1|)}. \quad \square$$

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 7.1. *Let A be a real $n \times n$ symmetric matrix. Then*

$$\begin{aligned} \frac{1}{n} \max_{X \succeq 0, X_{ii}=1} \langle A, X \rangle &\leq \lambda_{\max}(A) \\ \frac{1}{n} \min_{X \succeq 0, X_{ii}=1} \langle A, X \rangle &\geq \lambda_{\min}(A) \end{aligned}$$

Proof. We prove the upper bound below. The proof of the lower bound is identical.

$$\begin{aligned} \frac{1}{n} \max_{X \succeq 0, X_{ii}=1} \langle A, X \rangle &\leq \frac{1}{n} \max_{X \succeq 0, \text{tr}(X)=n} \langle A, X \rangle \\ &= \max_{X \succeq 0, \text{tr}(X)=1} \langle A, X \rangle \\ &= \lambda_{\max}(A). \end{aligned}$$

□

Recall $\alpha_{\text{gr}} := (c-1)(-\lambda_1\lambda_2)$ and $r_X := 2\sqrt{\alpha_{\text{gr}}}$.

Theorem 7.2. *Let $\mathcal{A} = (A_1, \dots, A_c)$ be a sequence of r -vertex atoms with edge weights ± 1 . Let \mathcal{H}_n denote a random n -lifted constraint graph and $\mathcal{I}_n = \mathcal{A}(\mathcal{H}_n)$ an associated instance graph with 1-wise uniform negations $(\xi_{ii'}^f)$. Let A_n be the adjacency matrix of \mathcal{I}_n . Then, with probability $1 - o_n(1)$,*

$$\begin{aligned} \max_{X \succeq 0, X_{ii}=1} \langle A_n, X \rangle &= (\lambda_1 + \lambda_2 + r_X \pm \varepsilon)n \\ \min_{X \succeq 0, X_{ii}=1} \langle A_n, X \rangle &= (\lambda_1 + \lambda_2 - r_X \pm \varepsilon)n. \end{aligned}$$

Proof. $\max_{X \succeq 0, X_{ii}=1} \langle A_n, X \rangle \geq (\lambda_1 + \lambda_2 + r_X - \varepsilon)n$ follows from [Theorem 5.2](#) and $\max_{X \succeq 0, X_{ii}=1} \langle A_n, X \rangle \leq (\lambda_1 + \lambda_2 + r_X + \varepsilon)n$ follows from [Fact 7.1](#). The upper and lower bounds on $\min_{X \succeq 0, X_{ii}=1} \langle A_n, X \rangle$ can be determined identically. □

Acknowledgments

We thank Yuval Peled for emphasizing the bipartite graph view of additive lifts, and Tselil Schramm for helpful discussions surrounding the trace method on graphs. S.M. would like to thank Jess Banks and Prasad Raghavendra for plenty of helpful discussions on orthogonal polynomials and nonbacktracking walks. Finally, we are grateful to Xinyu Wu for bringing the relevance of [\[BC18\]](#) to our attention and helping us to understand the issues discussed in [Section 1.3](#).

References

- [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. [12](#)

- [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. [1](#)
- [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. [11](#)
- [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. [16](#), [26](#)
- [AHL02] Noga Alon, Shlomo Hoory, and Nathan Linial. The Moore bound for irregular graphs. Graphs and Combinatorics, 18(1):53–57, 2002. [39](#)
- [Amb06] Andris Ambainis. Polynomial degree vs. quantum query complexity. Journal of Computer and System Sciences, 72(2):220–238, 2006. [11](#)
- [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. [11](#)
- [BC18] Charles Bordenave and Benoît Collins. Eigenvalues of random lifts and polynomial of random permutations matrices. arXiv preprint arXiv:1801.00876, 2018. [8](#), [9](#), [45](#)
- [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. [34](#)
- [Bel64] John Bell. On the Einstein Podolsky Rosen paradox. Physics Physique Fizika, 1(3):195–200, 1964. [11](#)
- [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. [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. [2](#)
- [BL06] Yonatan Bilu and Nathan Linial. Lifts, discrepancy and nearly optimal spectral gap. Combinatorica. An International Journal on Combinatorics and the Theory of Computing, 26(5):495–519, 2006. [14](#)
- [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. [6](#)

- [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. [10](#)
- [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](#), [34](#)
- [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. [2](#)
- [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. [11](#)
- [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. [11](#)
- [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. [10](#)
- [DMO⁺19] Yash Deshpande, Andrea Montanari, Ryan O’Donnell, Tselil Schramm, and Subhabrata Sen. The threshold for SDP-refutation of random regular NAE-3SAT. In Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 2305–2321, 2019. [2](#), [12](#), [32](#), [34](#), [38](#)
- [DMS17] Amir Dembo, Andrea Montanari, and Subhabrata Sen. Extremal cuts of sparse random graphs. The Annals of Probability, 45(2):1190–1217, 2017. [1](#)
- [DP93] Charles Delorme and Svatopluk Poljak. Laplacian eigenvalues and the maximum cut problem. Mathematical Programming, 62(1–3):557–574, 1993. [2](#), [10](#)
- [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. [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. [1](#)
- [Elo09] Yehonatan Elon. Gaussian waves on the regular tree. Technical Report 0907.5065, arXiv, 2009. [2](#)
- [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. [1](#), [2](#)

- [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](#), [5](#)
- [FS02] Uriel Feige and Gideon Schechtman. On the optimality of the random hyperplane rounding technique for Max-Cut. Random Structures and Algorithms, 20(3):403–440, 2002. [10](#)
- [GLS88] Martin Grötschel, László Lovász, and Alexander Schrijver. Geometric Algorithms and Combinatorial Optimization. Springer–Verlag, 1988. [10](#)
- [Gol00] Oded Goldreich. Candidate one-way functions based on expander graphs. Technical Report 90, Electronic Colloquium on Computational Complexity, 2000. [1](#)
- [Gro53] Alexander Grothendieck. Résumé de la théorie métrique des produits tensoriels topologiques. Boletín de la Sociedad Matemática São Paulo, 8:1–79, 1953. [10](#)
- [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. [2](#), [10](#)
- [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. [6](#)
- [HV15] Viktor Harangi and Bálint Virág. Independence ratio and random eigenvectors in transitive graphs. The Annals of Probability, 43(5):2810–2840, 2015. [2](#)
- [JP00] Ari Juels and Marcus Peinado. Hiding cliques for cryptographic security. Designs, Codes and Cryptography, 20(3):269–280, 2000. [1](#)
- [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. [6](#)
- [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. [1](#), [2](#)
- [Kub12] Carlos S Kubrusly. Spectral theory of operators on Hilbert spaces. Springer Science & Business Media, 2012. [16](#)
- [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. [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. [10](#)

- [Lyo17] Russell Lyons. Factors of IID on trees. Combinatorics, Probability and Computing, 26(2):285–300, 2017. 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. 6
- [MM09] Marc Mézard and Andrea Montanari. Information, physics, and computation. Oxford University Press, 2009. 1
- [MNS18] Elchanan Mossel, Joe Neeman, and Allan Sly. A proof of the block model threshold conjecture. Combinatorica, 38(3):665–708, 2018. 6
- [MO18] Sidhant Mohanty and Ryan O’Donnell. X-Ramanujan graphs, 2018. Available at <https://arxiv.org/abs/1904.03500>. 5, 6, 14
- [Mon17] Andrea Montanari. Bounds on ground state energy in the Sherrington–Kirkpatrick model, 2017. Open problem from AIM workshop, available at <http://aimpl.org/phaserandom/1/>. 2
- [Mon18] Andrea Montanari. Optimization of the Sherrington–Kirkpatrick hamiltonian. arXiv preprint arXiv:1812.10897, 2018. 2
- [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. 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. 2
- [Nor97] Sam Northshield. Several Proofs of Ihara’s theorem. 1997. 17
- [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. 11
- [Ram15] Farzaneh Ramezani. On the signed graphs with two distinct eigenvalues. arXiv preprint arXiv:1511.03511, 2015. 12
- [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. 1
- [Sen18] Subhabrata Sen. Optimization on sparse random hypergraphs and spin glasses. Random Structures & Algorithms, 53(3):504–536, 2018. 1
- [Tal06] Michel Talagrand. The Parisi formula. Annals of Mathematics. Second Series, 163(1):221–263, 2006. 1

- [Tsi80] Boris Tsirelson. Quantum generalizations of Bell's inequality. Letters in Mathematical Physics, 4(2):93–100, 1980. [11](#)
- [Wil07] Ryan Williams. Algorithms and resource requirements for fundamental problems. PhD thesis, Carnegie Mellon University, 2007. [11](#)
- [Woe00] Wolfgang Woess. Random walks on infinite graphs and groups, volume 138. Cambridge university press, 2000. [9](#)