

Computing the Independence Polynomial: from the Tree Threshold down to the Roots

Nicholas J. A. Harvey*

Piyush Srivastava[†]

Jan Vondrák[‡]

Abstract

We study an algorithm for approximating the multivariate independence polynomial $Z(\mathbf{z})$, with negative and complex arguments. While the focus so far has been mostly on computing combinatorial polynomials restricted to the univariate positive setting (with seminal results for the independence polynomial by Weitz (2006) and Sly (2010)), the independence polynomial with *negative* or *complex* arguments has strong connections to combinatorics and to statistical physics. The independence polynomial with negative arguments, $Z(-\mathbf{p})$, determines the *Shearer region*, the maximal region of probabilities to which the Lovász Local Lemma (LLL) can be extended (Shearer 1985). In statistical physics, complex zeros of the independence polynomial relate to existence of phase transitions.

Our main result is a deterministic algorithm to compute approximately the independence polynomial in any root-free complex polydisc centered at the origin. More precisely, we can $(1 + \varepsilon)$ -approximate the independence polynomial $Z(\mathbf{z})$ for an n -vertex graph of degree at most d , for any complex vector \mathbf{z} such that $Z(\mathbf{z}') \neq 0$ for $|z'_i| \leq (1 + \alpha)|z_i|$, in running time $(\frac{n}{\varepsilon\alpha})^{O(\log(d)/\sqrt{\alpha})}$. Our result also extends to graphs of unbounded degree that have a bounded connective constant. Our algorithm is essentially the same as Weitz’s algorithm for positive parameters up to the tree uniqueness threshold. The core of the analysis is a novel multivariate form of the correlation decay technique, which can handle non-uniform complex parameters. In summary, we provide a unifying algorithm for all known regions where $Z(\mathbf{z})$ is approximately computable. In particular, in the univariate real setting our work implies that Weitz’s algorithm works in an interval between two critical points $(-\lambda'_c(d), \lambda_c(d))$, and outside of this interval an approximation of $Z(\lambda)$ is known to be NP-hard.

As an application, we provide an algorithm to test membership in Shearer’s region within a multiplicative error of $1 + \alpha$, in running time $(n/\alpha)^{O(\sqrt{n/\alpha} \log d)}$. We also

give a deterministic algorithm for Shearer’s lemma (extending the LLL) with n events on m independent variables under slack α , with running time $(nm/\alpha)^{O(\sqrt{m/\alpha} \log d)}$.

On the hardness side, we prove that evaluating $Z(\mathbf{z})$ at an arbitrary point in Shearer’s region, and testing membership in Shearer’s region, are #P-hard problems. For Weitz’s correlation decay technique in the negative regime, we show that the $1/\sqrt{\alpha}$ dependence in the exponent is optimal.¹

1 Introduction

The *independence polynomial* is the generating function of independent sets of a graph. Formally, given a graph $G = (V, E)$, and a vector $\mathbf{x} = (x_v)_{v \in V}$ of *vertex activities*, it is the multi-linear polynomial

$$Z_G(\mathbf{x}) = \sum_{I \text{ indep. in } G} \prod_{v \in I} x_v.$$

Aside from its natural importance in combinatorics as a generating function, the independence polynomial has also been studied extensively in statistical physics where it arises as the *partition function* of the *hard core lattice gas*, which has been used as a model of adsorption. In both settings, the partition function and its derivatives encode important properties of the model. For example, in the combinatorial setting, Z_G encodes a weighted count of the independent sets, while the derivatives of $\log Z_G$ encode relevant average quantities, such as the mean size of an independent set. As such, much effort has gone into understanding the complexity of computing Z_G . The exact evaluation of the independence polynomial at non-trivial evaluation points turns out to be #P-hard [45]. As for approximate computation, the problem is well studied in the setting where the activities are *positive* and *real valued*. In this setting, the problem has served to highlight some of the tightest known connections between phase transitions and computational complexity: we will discuss this line of work in more detail below.

In this paper, we are concerned instead with the problem of approximately computing the independence poly-

*Email: nickhar@cs.ubc.ca. University of British Columbia.

[†]Email: piyush.srivastava@tifr.res.in. Tata Institute of Fundamental Research.

[‡]Email: jvondrak@stanford.edu. Stanford University.

¹An earlier version of this paper gave an algorithm with $1/\alpha$ dependence in the exponent, which would lead to trivial exponential running time in our applications.

nomial at possibly negative and even complex valued vertex activities. The interest in studying partition functions at complex values of the activities originally comes from statistical mechanics, where there is a paradigm of studying phase transitions in terms of the analyticity of $\log Z_G$. This paradigm has led to the question of characterizing regions of the complex plane where the partition function is non-zero [24, 47]. Inspired by previous connections between statistical physics and computation complexity, a natural question is: does the maximum radius around the origin within which $\log Z_G$ is analytic (i.e., within which Z_G has no roots) correspond to a transition in the computational complexity of computing Z_G ? As we discuss below, the answer is yes.

A second motivation for studying the independence polynomial at complex activities comes from a delightful connection between combinatorics and statistical mechanics that arose in the work of Shearer [35] and Scott and Sokal [34] on the Lovász Local Lemma (LLL). In particular, the largest region of parameters in which the LLL applies is the maximal connected region of the negative orthant within which Z_G has no roots. As we discuss below, an algorithm for approximating Z_G at negative activities has several algorithmic applications relating to the LLL, including testing whether the hypotheses are satisfied, as well as giving a constructive proof of the LLL itself.

1.1 Our results

Before stating our results, let us define the *Shearer region* for graph G to be

$$(1.1) \quad \mathcal{S} = \mathcal{S}_G := \left\{ \mathbf{p} \in [0, 1]^V : Z_G(\mathbf{z}) \neq 0 \quad \forall \mathbf{z} \in \mathbb{C}^V \text{ s.t. } |\mathbf{z}| \leq \mathbf{p} \right\},$$

which describes the radii of polydiscs within which Z_G has no roots. Here, $|\mathbf{z}|$ means coordinate-wise magnitude, and \leq also applies coordinate-wise. It can be shown that \mathcal{S}_G is an open set.

Our main result is a fully polynomial time approximation scheme (FPTAS) for $Z_G(\mathbf{z})$ when \mathbf{z} is a vector of possibly complex activities for which the vector $|\mathbf{z}|$ of their magnitudes lies in the Shearer region.

Theorem 1.1 (FPTAS for Z_G). Let G be an n -vertex graph with maximum degree d . Suppose that $\alpha, \varepsilon \in (0, 1]$, and that $\mathbf{z} \in \mathbb{C}^V$ satisfies $(1 + \alpha) \cdot |\mathbf{z}| \in \mathcal{S}_G$. Then a $(1 + \varepsilon)$ -approximation to $Z_G(\mathbf{z})$ can be computed in time $\left(\frac{n}{\varepsilon\alpha}\right)^{O(\log(d)/\sqrt{\alpha})}$.

As the set \mathcal{S}_G is somewhat mysterious, it is instructive to consider the following univariate corollary.

Corollary 1.2 (FPTAS for the univariate case). Let G be an n -vertex graph with maximum degree d . Define $\lambda_G = \min\{|\mathbf{z}| : \mathbf{z} \in \mathbb{C}, Z_G(\mathbf{z}\mathbf{1}) = 0\}$. Let $\alpha, \varepsilon \in (0, 1]$

and $z \in \mathbb{C}$ satisfy $(1 + \alpha)|z| \leq \lambda_G$. Then a $(1 + \varepsilon)$ -approximation to $Z_G(\mathbf{z}\mathbf{1})$ can be deterministically computed in time $\left(\frac{n}{\varepsilon\alpha}\right)^{O((1/\sqrt{\alpha}) \cdot \log(d))}$.

Remark: Region of applicability. In order to understand λ_G it is helpful to consider a bound that depends only on the degree d . Define $\lambda'_c(d)$ to be the minimum of λ_G over all graphs of maximum degree d . Then it is known [35] that $\lambda'_c(1) = 1/2$ and $\lambda'_c(d) = \frac{(d-1)^{d-1}}{d^d}$ for $d \geq 2$; the minimum is achieved by the infinite d -regular tree. So $\lambda'_c(d)$ is the threshold, depending only on d , that determines the region of applicability of our algorithm. This is no accident: approximating $Z_G(\mathbf{z}\mathbf{1})$ for real $z < -\lambda'_c$ has recently been shown to be NP-hard by Galanis, Goldberg and Štefankovič [14], showing that Corollary 1.2 has the tightest possible range of applicability on the negative real line (i.e., the Shearer region). Thus, a phase transition in the computational complexity of the problem occurs right at the boundary of the region within which Z_G is guaranteed to have no roots. As Figure 1 shows, we now have a complete picture of the computational complexity of Z_G in the real univariate case, as a function of d .

Remark: Dependence on Slack. An important feature of Theorem 1.1 is that although the running time degrades as the input vector \mathbf{p} approaches the boundary of the Shearer region \mathcal{S} , the degradation is only *sub-exponential* in $\frac{1}{\alpha}$ (being exponential in $\frac{1}{\sqrt{\alpha}}$) where α is the *slack* parameter that measures the distance to the boundary. This is in contrast to an earlier manuscript of the present paper [19] and the concurrent paper of Patel and Regts [31], which (using different methods), obtained an FPTAS whose running time is actually *exponential* in $\frac{1}{\alpha}$. We describe the new ideas required to get this better dependence on α in Section 3.2, and remark on the barriers to improving this dependence towards the end of this subsection.

The importance of a sub-exponential dependence on the slack is that for some applications it is imperative to approximate the independence polynomial at points that are extremely close to the boundary of the Shearer region and have slack at most $\Theta(1/n)$. We present two such applications here, for both of which we are able to obtain sub-exponential time algorithms, and for both of which the earlier results [19,31] only give exponential time algorithms.

Remark: Connective constant. Theorem 1.1 extends to graphs of unbounded maximum degree that have a bounded *connective constant* [17, 28, 37, 38]. See the full version for the details of this extension.

Application 1: Testing membership in Shearer region. Physicists have studied the univariate threshold λ_G for specific graphs, as this determines the region within which there are no phase transitions [47]. For example, to under-

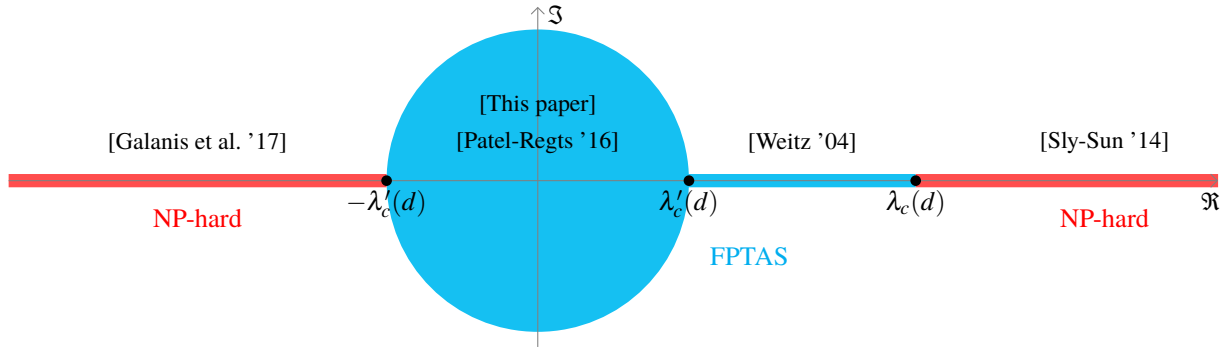


Figure 1: Summary of results for computation of $Z_G(\mathbf{z}\mathbf{1})$ in the complex univariate setting, as a function of the degree d . Here $\lambda'_c = \frac{(d-1)^{d-1}}{d^d} \searrow \frac{1}{ed}$ and $\lambda_c = \frac{(d-1)^{d-1}}{(d-2)^d} \searrow \frac{e}{d}$. Note that a major difference of this work from Patel-Regs [31] that is not captured by this figure is that our running time has a significantly better dependence on distance from the boundary of Shearer’s region, which is crucial in our applications.

stand phase transitions in \mathbb{Z}^2 , researchers have performed numerical computations on finite graphs to estimate the exact value $\lambda_G(\mathbb{Z}^2)$. (See, e.g., [22] [34, Section 8.4] [44].) Computations have shown that $\lambda_G(\mathbb{Z}^2) \leq 1/8$ (rigorous) and $\lambda_G(\mathbb{Z}^2) = 0.119, 338, 881, 88(1)$ (non-rigorous).

Our first application is an algorithm to test whether a given vector \mathbf{p} lies in the Shearer region, up to accuracy α . This can be used to compute bounds on λ_G , and could potentially be useful for physicists.

Theorem 1.3. Given a graph G , $\mathbf{p} \in (0, 1)^V$, and $\alpha \in (0, 1]$, there exists a deterministic algorithm which, in running time $(n/\alpha)^{O(\sqrt{n/\alpha \log d})}$ decides whether $\mathbf{p} \in \mathcal{S}_G$ or $(1 + \alpha)\mathbf{p} \notin \mathcal{S}_G$.

This algorithm uses the FPTAS of Theorem 1.1 in a black box fashion, calling it at points in \mathcal{S} that may have slack $O(1/n)$. Replacing the black box by an algorithm that had an exponential dependence on the slack would give an algorithm with only a trivial exponential time

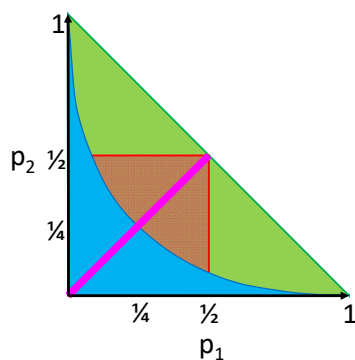


Figure 2: For illustrative purposes, let us consider the graph $G = K_2$; here $d = 1$ and $\lambda'_c(d) = 1/2$. Our algorithm applies throughout Shearer’s region \mathcal{S}_G , shown as the green triangular region. The pink line segment is the restriction of \mathcal{S}_G for the univariate function $Z_G(\mathbf{z}\mathbf{1})$. The work of Patel-Regs describes an algorithm focused on the univariate case, but they mention [31, pp. 13] that it can be generalized to all points dominated by λ'_c (the red square region). The blue region, defined as \mathcal{L} in Appendix A, is where the original LLL [12, 41] applies.

guarantee on its run-time. We note also that the testing membership in \mathcal{S} is #P-hard when α is exponentially small. (See Appendix E for a precise statement of this hardness result.)

Application 2: Constructive algorithm for the Lovász Local Lemma by polynomial evaluation. The LLL is a tool in combinatorics giving conditions ensuring that it is possible to avoid certain bad events $\mathcal{E}_1, \dots, \mathcal{E}_n$. (For readers unfamiliar with the LLL, a statement is provided in Appendix A.) Although the LLL guarantees that there exists a point in $\bigcap_{i=1}^n \overline{\mathcal{E}_i}$, it provides no hint on how to find such a point. For decades, algorithmically constructing such a point was a major research challenge, though over the past 10 years dramatic progress has been made. Any such algorithm must necessarily make some assumptions on the probability space, the most common being the “variable model” used by [29]. All previous algorithms have been based on the idea of randomly sampling variables followed by brute-force search [2, 8], or random resampling [1, 18, 20, 23, 29, 30, 42], or derandomizations of those ideas [2, 8, 10, 29, 30].

We develop a completely new algorithmic approach to the LLL in the variable model. The previous randomized algorithms can be viewed as generating a sequence of *infeasible, integral* solutions; at each step, they resample one of the bad events and hopefully move closer to feasibility. (The previous deterministic algorithms are derandomizations of this approach.) In contrast, our new algorithm generates a sequence of *feasible, fractional* solutions; at each step, it fixes the value of one of the variables while preserving feasibility in \mathcal{S}_G . The value of the polynomial $Z_G(\mathbf{z})$ is used to determine membership in \mathcal{S}_G . Thus, our algorithm can be viewed as a *rounding algorithm* for the LLL, and the value of $Z_G(\mathbf{z})$ can be viewed as a *pessimistic estimator* for the probability of $\bigcap_{i=1}^n \overline{\mathcal{E}_i}$. To compute $Z_G(\mathbf{z})$, our algorithm uses (as a black box) our deterministic FPTAS for evaluating the independence polynomial with negative activities and slack $\Omega(1/m)$, where m is the number of variables.

Theorem 1.4. Consider an LLL scenario in the variable model (as in Appendix A.1): $\mu_{\mathbf{z}}$ is the product distribution on $\{0, 1\}^m$ with expectation \mathbf{z} , G is the dependency graph for events $\mathcal{E}_1, \dots, \mathcal{E}_n$, and $p_i = \mu_{\mathbf{z}}(\mathcal{E}_i)$. There is a deterministic algorithm that takes as input a description of the events $\mathcal{E}_1, \dots, \mathcal{E}_n$, a vector $\mathbf{z} \in [0, 1]^m$, and a parameter $\alpha \in (0, 1]$ such that $(1 + \alpha) \cdot \mathbf{p}(\mathbf{z}) \in \mathcal{S}_G$. The algorithm runs for time $(nm/\alpha)^{O(\log(d)\sqrt{m/\alpha})}$ and outputs a point in $\bigcap_{i=1}^n \overline{\mathcal{E}_i}$.

This algorithm uses our FPTAS from Theorem 1.1 as a black box. Note that our algorithm runs in subexponential time, so, as of now, its runtime is not competitive with the state of the art deterministic algorithms for the LLL [10]. Nevertheless, prior to our work there was essentially only one known algorithmic technique known for the LLL: the witness tree technique originating with Beck [8]. Our work provides the only other known technique that gives an algorithm for the LLL better than brute-force. The fact that our algorithm is slow is only because the best known implementation of the black box (i.e., Theorem 1.1) has a running time that depends sub-exponentially on the slack. The algorithm thus points to a new intriguing connection between approximate counting and algorithmic versions of the LLL, and suggests the open question of finding the optimal dependence on the slack $\sqrt{\alpha}$ in Theorem 1.1.

Dependence on the “slack parameter” α . The discussion following the two applications above suggests that the question of the optimal dependence on the slack α in Theorem 1.1 is of importance for further exploration of the connection between approximate counting and the LLL. While we cannot yet provide a complete answer to this question, we conclude this section with a couple of our results that address this point. Our first result in this direction shows that some dependence on the slack parameter is inevitable. (See Appendix E.1 for a proof).

Theorem 1.5 (Necessity of slack). If there is an algorithm to estimate $Z_G(-\mathbf{p})$, assuming $(1 + \alpha)\mathbf{p} \in \mathcal{S}$, within a $\text{poly}(n)$ multiplicative factor in running time $(n \log \frac{1}{\alpha})^{O(\log n)}$ then $\#P \subseteq \text{DTIME}(n^{O(\log n)})$.

However, this hardness result, while applying to all algorithmic approaches, only provides a weak lower bound on what can be achieved. We do not yet have any stronger general lower bounds, but our second result, described in detail in Appendix F, presents evidence that the dependence on $1/\sqrt{\alpha}$ in Theorem 1.1 is optimal for the techniques used in our paper. Nevertheless, it does not preclude the possibility that other approximate counting techniques could substantially improve upon Theorem 1.1. We discuss some related future directions in Section 6.

1.2 Related work

As discussed above, the exact computation of the independence polynomial turns out to be #P-hard. This is a fate shared by the partition functions of several other “spin systems” (e.g., the Ising model) in statistical physics, and by now there is extensive work on the complexity theoretic classification of partition functions in terms of dichotomy theorems (see e.g., [9]).

The approximation problem for a univariate partition function with a positive real argument is also well studied and has strong connections with phase transitions in statistical mechanics. In two seminal papers, Weitz [46] and Sly [39] (see also [13, 15, 40]) showed that there exists a critical value $\lambda_c(d)$ such when $\lambda < \lambda_c(d)$, there is an FPTAS for the partition function $Z_G(\lambda)$ on graphs of maximum degree d , while for $\lambda > \lambda_c(d)$ close to the threshold, approximating $Z_G(\lambda)$ on d -regular graphs is NP-hard under randomized reductions. (Sly and Sun [40] extended the hardness result to any $\lambda > \lambda_c(d)$.)

The approach for our FPTAS builds upon the correlation decay technique pioneered by Weitz, which has since inspired several results in approximate counting (see, e.g., [7, 11, 16, 25–27, 36, 37]). Unlike previous work, where the partition function has positive activities and induces a probability distribution on the underlying structures, our emphasis is on negative and complex activities. It turns out that Weitz’s proof can be easily modified to handle a *univariate* independence polynomial with a *negative* (and indeed, *complex*) parameter z satisfying $|z| < \lambda'_c(d) = \frac{(d-1)^{d-1}}{d^d}$, analogous to the $\lambda < \lambda_c(d)$ condition mentioned above; this observation appears in [43]. Our work considers a much more general scenario: the *multivariate* independence polynomial under a global condition incorporating all vertex activities (i.e., the set \mathcal{S}). This yields a result for the univariate case *stronger* than [43], as our threshold λ_G in Corollary 1.2 depends on G not just on d .

Starting with a paper of Barvinok [4], a different approach to approximating partition functions in their zero-free regions has emerged. Here, the analyticity of $\log Z$ in the zero-free region of Z is used to provide an additive approximation to $\log Z$ (which translates to a multiplicative approximation for Z) via a Taylor expansion truncated at an appropriate degree. While this method has by now been applied to several classes of partition functions [3, 5, 6, 32], the resulting algorithms had turned out to be quasi-polynomial in the earlier applications because of the lack of a method to efficiently compute coefficients of terms of degree $\Omega(\log n)$ in the Taylor expansion of $\log Z$ (which, in the case of the hard core model, correspond to $\Omega(\log n)$ -wise correlations among vertices in a random independent set). In work that was circulated concurrently with an earlier manuscript [19] of this paper, Patel and

Regts [31] showed that for a class of models, these coefficients could be computed in polynomial time on bounded degree graphs, and as a consequence obtained an FPTAS for some partition functions in the region of analyticity of their logarithms. This included the *univariate* independence polynomial in bounded-degree graphs of degree d when the activity λ satisfies $|\lambda| < \lambda'_c$. While their particular result for the univariate independence polynomial seems to be implied by the observations in [43] pointed out above, their technique applies also to other models. We note however that the present work has advantages over the result of Patel and Regts in two qualitative aspects which are both crucial for our applications.

First, as mentioned above, the running time of our FPTAS is sub-exponential in $1/\alpha$ when the input activity (or more generally, the input probability vector in the multivariate case) has slack α , whereas their algorithm has an exponential dependence on $1/\alpha$. Indeed, it appears that this exponential dependence on $1/\alpha$ is intrinsic to their approach since the rate of convergence of the power series they use for approximating $\log Z$ is exactly $1 - \alpha$ for an activity that has slack α , so that the number of terms of the series that need to be evaluated for an additive ε/n -approximation to $\log Z$ (which corresponds to a $(1 \pm \Theta(\varepsilon/n))$ multiplicative approximation for Z) is $\Omega(\log_{1/(1-\alpha)} \log(n/\varepsilon)) = \Omega\left(\frac{\log(n/\varepsilon)}{\alpha}\right)$. Since the complexity of computing the k th term of this series in their framework is $\Omega(d^k)$, this leads to a run time that is $(n/\varepsilon)^{\Omega((1/\alpha)\cdot\log(d))}$ (our algorithm, in contrast will run in time $(n/\alpha\varepsilon)^{O((1/\sqrt{\alpha})\cdot\log(d))}$). As discussed above, this improvement over Patel and Regts [31] is crucial for the applications considered in this paper. Second, our paper explicitly handles the multivariate independence polynomial. The work of Patel-Regts describes an algorithm focused on the univariate case, but they mention [31, pp. 13] that it can be generalized to all points dominated by λ'_c (the red square region in Figure 2). Though it seems plausible that their method can be extended to be applicable throughout the Shearer region (albeit still with an exponential dependence on the slack α), to the best of our knowledge, the algorithmic details for doing so have not yet been published.

1.3 Techniques

As in previous work, our starting point is the standard self-reducibility argument showing that the problem of designing an FPTAS for $Z_G(\lambda)$ is equivalent to the problem of designing an FPTAS for computing the *occupation ratio* r_v of a given vertex v in any given graph G (i.e., the ratio of the total weights of the independent sets containing v to the total weight of those that do not). In previous work, these occupation ratios are actual likelihood ratios that can be translated to the probability that the vertex v is occu-

pied, but because of complex weights, we do not have the luxury of this interpretation. However, as in earlier work, we can still write formal recurrences for these occupation ratios. As Weitz showed [46], this recursive computation is naturally structured as a tree which has the same structure as the tree $T_{\text{SAW}}(v, G)$ whose nodes correspond to self-avoiding walks in G starting at v .² However the tree $T_{\text{SAW}}(v, G)$ has size exponential in $|V|$, so this reduction does not immediately give a polynomial time algorithm.

The crucial step in correlation decay algorithms is to show that this tree can be truncated to polynomial size without incurring a large error in the value computed at the root. In earlier work on positive activities, especially since Restrepo et al. [33], the standard method for doing this has been to consider instead a recurrence for an appropriately chosen function $\phi(r_v)$, known as the *message*, that is chosen so that when correlation decay holds on the d -regular tree, each step of the recurrence on the truncated T_{SAW} contracts the error introduced by the truncation by a constant factor. Thus, by expanding the tree to $\ell = O(\log \frac{n}{\varepsilon})$ levels (so $d^\ell = \text{poly}(n, \frac{1}{\varepsilon})$ nodes), one obtains a $(1 + O(\frac{\varepsilon}{n}))$ -approximation to the value at the root.

Our approach also involves truncating the computation tree at an appropriate depth and then controlling the errors introduced due to truncation. However, in part because of the lack of a uniform bound on the vertex activities, we are not able to recreate a message-based approach. Instead, we perform a direct amortization argument, where we define recursively for each node in the computation tree an *error sensitivity parameter*, and then measure errors at that node as a fraction of the local error sensitivity parameter. We then establish two facts: (1) that the errors, when measured as a fraction of the error sensitivity parameter, do indeed decay by a constant fraction (roughly $(1 - \Omega(\sqrt{\alpha}))$) when the input probability vector has slack α at each step of the recurrence (even though the absolute errors may not), and (2) that the error sensitivity parameter of the root node is not too large, so that the absolute error of the final answer can be appropriately bounded. The detailed argument appears in Section 3.2. For readers familiar with the earlier manuscript [19] of this paper, we point out that in that manuscript, the decay at each step of the recurrence could only be shown to be of the form $(1 - O(\alpha))$; informal and formal descriptions of how this is improved to $(1 - \Omega(\sqrt{\alpha}))$ in the present paper also appear in Section 3.2.

2 Overview of the correlation decay method

In this section we summarize the basic concepts and facts relating to Weitz's correlation decay method. Since all

² Interpreting the computation tree as the T_{SAW} tree will have less prominence in our analysis than in previous work.

the claims are simple or known, the proofs are omitted or appear in the appendix.

Partition functions and occupation ratios. Since we are primarily interested in the hard-core partition function (i.e., independence polynomial) with *negative* activities, it will be convenient to introduce the following notation. Let $G = (V, E)$ be a fixed graph, and let \mathbf{p} be a fixed vector of (possibly complex) parameters on the vertices of V . For $S \subseteq V$, let $\text{Ind}(S) = \text{Ind}_G(S) = \{I \subseteq S : I \text{ independent in } G\}$. Following the notation of [21, 23], we define the *alternating-sign independence polynomial* for any subset S of V to be

$$\check{q}_S = \check{q}_S(\mathbf{p}) := \sum_{I \in \text{Ind}(S)} (-1)^{|I|} \prod_{v \in I} p_v.$$

Note that $\check{q}_V(\mathbf{p}) = Z_G(-\mathbf{p})$. The computation of \check{q}_V will be reduced to the computation of *occupation ratios* defined as follows. For a pair (S, u) , where $S \subseteq V$ and $u \in S$, the occupation ratio $r_{S,u}$ is

$$(2.1) \quad r_{S,u} = r_{S,u}(\mathbf{p}) := -\frac{\sum_{I \in \text{Ind}(S), u \in I} (-1)^{|I|} \prod_{v \in I} p_v}{\sum_{I \in \text{Ind}(S), u \notin I} (-1)^{|I|} \prod_{v \in I} p_v}.$$

For readers familiar with the notation of Weitz [46], we note that $r_{S,u}$ agrees with his definition of occupation ratios except for the negative signs used in the definition here. Using the definition of \check{q}_S , and the notation $\Gamma(u) = \{v : u \text{ is a neighbor of } v \text{ in } G\}$, $\Gamma^+(u) = \Gamma(u) \cup \{u\}$, we can also rewrite this quantity as

$$(2.2) \quad r_{S,u} = \frac{p_u \check{q}_{S \setminus \Gamma^+(u)}}{\check{q}_{S \setminus \{u\}}} = -\frac{\check{q}_S - \check{q}_{S \setminus \{u\}}}{\check{q}_{S \setminus \{u\}}} = 1 - \frac{\check{q}_S}{\check{q}_{S \setminus \{u\}}}.$$

A standard self-reducibility argument now reduces the computation of \check{q}_V to that of the $r_{S,u}$.

Claim 2.1. Fix an arbitrary ordering (v_1, v_2, \dots, v_n) of V , and let $S_i = \{v_i, v_{i+1}, \dots, v_n\}$. We then have

$$(2.3) \quad \check{q}_V = \prod_{i=1}^n \frac{\check{q}_{S_i}}{\check{q}_{S_{i+1}}} = \prod_{i=1}^n (1 - r_{S_i, v_i}).$$

Recurrences for the occupation ratios. An important observation in Weitz's work [46] was that the computation of occupation ratios similar to the $r_{S,u}$ can be carried out over a tree-like recursive structure. We follow a similar strategy, although we find it convenient to work with a somewhat different notation.

Definition 2.2 (Child subproblems). Given a pair (S, u) with $S \subseteq V$ and $u \in S$, and an arbitrary ordering (v_1, v_2, \dots, v_k) of $\Gamma(u) \cap S$, we define the set of *child subproblems* $\mathcal{C}(S, u) = \mathcal{C}_{(v_1, v_2, \dots, v_k)}(S, u)$ to be

$$\mathcal{C}(S, u) = \{(S \setminus \{u\}, v_1), (S \setminus \{u, v_1\}, v_2), \dots, (S \setminus \{u, v_1, \dots, v_{k-1}\}, v_k)\}.$$

Note that the ordering of neighbors used in the definition of $\mathcal{C}(S, u)$ is completely arbitrary and orderings between neighbors of different vertices do not share any consistency constraints. The recurrence relation for the computation of the $r_{S,u}$, analogous to Weitz's computation tree, is then the following:

Lemma 2.3 (Computational recurrence). Fix a graph $G = (V, E)$ and a vector $\mathbf{p} \in \mathbb{C}^V$ of complex parameters. Let (S, u) be such that $u \in S$ and $S \subseteq V$. Fix an arbitrary ordering (v_1, v_2, \dots, v_k) of $\Gamma(u) \cap S$ and define the corresponding set $\mathcal{C}(S, u)$ of child subproblems. We then have

$$(2.4) \quad r_{S,u} = p_u \cdot \prod_{(S', u') \in \mathcal{C}(S, u)} \frac{1}{1 - r_{S', u'}}.$$

Remark 2.1. As observed by Weitz [46], each node in the computation tree of $r_{V,v}$ at depth ℓ corresponds to a unique self-avoiding walk of length ℓ starting at v .

The Shearer region. The Shearer region was defined (implicitly) by Shearer [35] as follows.

Definition 2.4 (Shearer region and slack). Given a graph $G = (V, E)$, the *Shearer region* \mathcal{S} is the set of vectors $\mathbf{p} \in (0, 1)^V$ such that $\check{q}_S(\mathbf{p}) > 0$ for all $S \subseteq V$. A probability vector \mathbf{p} is said to have *slack* $\alpha \geq 0$ if the vector $(1 + \alpha)\mathbf{p}$ is also a probability vector and is contained in \mathcal{S} .

Shearer proved that this is the maximal region of probability vectors to which the LLL can be extended. The equivalence with our earlier definition (1.1) is due to Scott and Sokal [34, Theorem 2.10]: the Shearer region can be equivalently defined by the absence of roots in a certain polydisc, as follows.

Theorem 2.5. A probability vector $\mathbf{p} \in (0, 1)^V$ is in the Shearer region (as defined in Definition 2.4) if and only if for all vectors $\mathbf{z} = (z_v)_{v \in V}$ of complex activities such that $|z_v| \leq p_v$, it holds that $Z_G(\mathbf{z}) \neq 0$.

Considering this, it is natural to extend the definition of the Shearer region to complex parameters. In the following, we consider primarily this complex extension of the Shearer region.

Definition 2.6 (Complex Shearer region). Given a graph $G = (V, E)$, the *complex Shearer region* is

$$\mathcal{S} = \mathcal{S}_G := \{\mathbf{p} \in \mathbb{C}^V : Z_G(\mathbf{z}) \neq 0 \ \forall \mathbf{z} \in \mathbb{C}^V, |z_v| \leq |p_v|\}.$$

We now state some important properties of the occupation ratios and \check{q}_S in the setting of real, positive parameters \mathbf{p} . These results are essentially translations of the results of [34, 35] into our notation.

Lemma 2.7 (Monotonicity and positivity of \check{q}). Let $G = (V, E)$ be any graph and let $\mathbf{p} \in (0, 1)^V$ be such that $\mathbf{p} \in \mathcal{S}$. Then for any subsets A and B of V such that $A \subseteq B$,

Algorithm 1 Our algorithm to compute a $(1 + \varepsilon)$ -approximation to $\check{q}_V(\mathbf{p})$.

```

1: procedure COMPUTEINDEPPOLY( $G = (V, E), \mathbf{p}, \ell$ )
2:   Fix an ordering  $V = (v_1, v_2, \dots, v_n)$ 
3:    $\check{q} \leftarrow 1$ 
4:   for  $i \leftarrow 1, \dots, n$  do
5:      $\check{q} \leftarrow \check{q} \cdot (1 - \text{OccRATIO}(G, \mathbf{p}, \ell, \{v_i, \dots, v_n\}, v_i))$ 
6:   return  $\check{q}$ 
7: procedure OccRATIO( $G, \mathbf{p}, \ell, S, u$ )
8:   if  $\ell = 0$  then return 0
9:   Let  $(w_1, \dots, w_k)$  be a fixed ordering of  $S \cap \Gamma(u)$ 
10:   $r \leftarrow p_u$ 
11:  for  $i \leftarrow 1, \dots, k$  do
12:     $r \leftarrow \frac{r}{1 - \text{OccRATIO}(G, \mathbf{p}, \ell - 1, S \setminus \{u, w_1, \dots, w_{i-1}\}, w_i)}$ 
13:  return  $r$ 

```

we have $\check{q}_A(\mathbf{p}) \geq \check{q}_B(\mathbf{p}) > 0$.

Lemma 2.8 (Occupation ratios are bounded). Let $G = (V, E)$ be any graph and let $\mathbf{p} \in (0, 1)^V$ be such that $\mathbf{p} \in \mathcal{S}$. Then, for any subset S of V and any vertex $u \in S$, we have $p_u \leq r_{S,u} < 1$.

The correlation decay algorithm. Weitz’s high-level approach to compute the independence polynomial is to compute the partition function via a telescoping product analogous to (2.3), and to compute each occupation ratio via a recurrence analogous to (2.4). As discussed in Section 1.3, the recursion is truncated to ℓ levels, and the analysis shows that the occupation ratio at the root is not affected heavily by the occupation ratios where the truncation occurred.

We follow that same high-level approach here, although the details of the analysis are quite different. Algorithm 1 presents pseudocode giving a compact description of the full algorithm. The main procedure, COMPUTEINDEPPOLY(G, \mathbf{p}, ℓ) implements (2.3) to estimate $\check{q}_V(\mathbf{p})$ for a graph $G = (V, E)$, a parameter vector \mathbf{p} , and a desired recursion depth ℓ . (The required value of ℓ depends upon the the accuracy parameter ε ; see Theorem 3.9). The recursive procedure OccRATIO($G, \mathbf{p}, \ell, S, u$) implements (2.4) to estimate the occupation ratio $r_{S,u}$ by executing ℓ levels of recursion.

3 The analysis

Let us turn to the analysis of the correlation decay method in our setting. The notion of *correlation decay* in the hard core model refers to the decaying dependence of the occupation probability at a given vertex v on the conditioning on a set of vertices at a certain distance from v . In the setting of positive activities \mathbf{z} [46], these correlations are closely tied to the decay of errors in the computation tree for $r_{S,u}$ described in (2.4). For negative or general

complex activities, the occupation ratios $r_{S,u}$ do not have a direct interpretation in terms of occupation probabilities. However, the analysis of errors in the computation tree is reminiscent of that of [46] and hence we still refer to it as *correlation decay*.

Unlike Weitz’s setting [46], where all vertex activities are the same, and the bounds are derived uniformly for all graphs with degrees bounded by d , here we are aiming for a more refined analysis for a particular graph G and a (possibly non-uniform) vector \mathbf{p} . In Weitz’s setting, the worst-case errors in the recursive tree can be proved to decay in a uniform fashion (possibly after an application of an appropriate potential function or message). That is not the case here, since the local structure of G and \mathbf{p} might cause the errors to locally increase, even if the computation eventually converges. Hence it is critical to identify a local sensitivity parameter that describes how the errors propagate in the recursive tree.

3.1 The error sensitivity parameter

For simplicity of notation, we fix the input graph $G = (V, E)$ and an ordering on vertices $V = \{v_1, \dots, v_n\}$ for the rest of this section. Recall that $\mathbf{p} = (p_1, p_2, \dots, p_n)$ denotes a vector of vertex parameters in the complex plane. (We have $\mathbf{p} = -\mathbf{z}$ where \mathbf{z} are the usual activities in the hard core model.) We use $|\mathbf{p}|$ to denote the vector $(|p_1|, |p_2|, \dots, |p_n|)$. Note that \mathbf{p} is in the Shearer region \mathcal{S} if and only if $|\mathbf{p}|$ is in \mathcal{S} .

First, let us consider how the errors propagate throughout the recursive computation in Algorithm 1. Let $R_{S,u}$ an estimate obtained by the algorithm for the occupation ratio $r_{S,u} = 1 - \check{q}_S / \check{q}_{S \setminus \{u\}}$. We are interested in how the additive approximation error $|r_{S,u} - R_{S,u}|$ propagates in the recursive computation.

It turns out that \mathbf{p} being real positive is in some sense the worst case; to simplify notation, we define for $u \in S$ and $\mathbf{p} \in \mathbb{C}^V$,

$$\rho_{S,u}(\mathbf{p}) := r_{S,u}(|\mathbf{p}|).$$

The reader who wishes to understand the main ideas while avoiding some mild technical details may henceforth assume that \mathbf{p} is a real positive vector, and therefore $\rho_{S,u}(\mathbf{p}) = r_{S,u}(\mathbf{p})$. Indeed, an easy recursive argument shows that when $\mathbf{p} \in \mathcal{S}$, $\rho_{S,u}(\mathbf{p})$ dominates both $r_{S,u}(\mathbf{p})$ and $R_{S,u}(\mathbf{p})$ (see Claim 3.6):

$$|r_{S,u}(\mathbf{p})| \leq \rho_{S,u}(\mathbf{p}) \text{ and } |R_{S,u}(\mathbf{p})| \leq \rho_{S,u}(\mathbf{p}).$$

Now, from the mean value theorem, we obtain the following recursive bound.

Claim 3.1. Let \mathbf{p} lie in the complex Shearer region. For a node (S, u) with children $\mathcal{C}(S, u)$ in the recursive

computation tree, we have

$$(3.1) \quad |r_{S,u} - R_{S,u}| \leq \rho_{S,u} \sum_{c \in \mathcal{C}(S,u)} \frac{|r_c - R_c|}{1 - \rho_c}.$$

Proof. Recall Lemma 2.3, $r_{S,u} = p_u \prod_{c \in \mathcal{C}(S,u)} \frac{1}{1 - r_c}$. Using the mean value theorem (Theorem C.3) with $\gamma_c := \rho_c$, we obtain

$$\begin{aligned} |r_{S,u} - R_{S,u}| &\leq |p_u| \prod_{c \in \mathcal{C}(S,u)} \frac{1}{1 - \rho_c} \sum_{c' \in \mathcal{C}(S,u)} \frac{|r_{c'} - R_{c'}|}{1 - \rho_{c'}} \\ &= \rho_{S,u} \sum_{c \in \mathcal{C}(S,u)} \frac{|r_c - R_c|}{1 - \rho_c}. \end{aligned}$$

Note that the conditions imposed in the hypothesis of Theorem C.3 hold, since, as pointed out above, $|r_{S,u}|, |R_{S,u}| \leq \rho_{S,u}$ for all nodes (S,u) in the computation tree (see Claim 3.6). \square

Our error sensitivity parameter is defined to capture how errors propagate under this recursive bound. An important observation is that the derivative of $\rho_{S,u}((1+t)\mathbf{p})$ with respect to t satisfies a recurrence very similar to Claim 3.1, and this is the main motivation behind the following definition.

Definition 3.2 (Error sensitivity parameter). The *error sensitivity parameter* $\beta_{S,u} := \beta_{S,u}(\mathbf{p})$ is defined as

$$(3.2) \quad \beta_{S,u}(\mathbf{p}) := \left. \frac{d\rho_{S,u}((1+t)\mathbf{p})}{dt} \right|_{t=0}.$$

Claim 3.3. Let (S,u) be node in the computation tree. Then

$$\beta_{S,u} = \rho_{S,u} \cdot \left(1 + \sum_{c \in \mathcal{C}(S,u)} \frac{\beta_c}{1 - \rho_c} \right).$$

Proof. By a direct calculation using the definition of $\beta_{S,u}$,

$$\begin{aligned} \beta_{S,u} &:= \left. \frac{d\rho_{S,u}((1+t)\mathbf{p})}{dt} \right|_{t=0} \\ &= \left. \frac{d}{dt} \left((1+t)|p_u| \cdot \prod_{c \in \mathcal{C}(S,u)} \frac{1}{1 - \rho_c((1+t)\mathbf{p})} \right) \right|_{t=0} \\ &= \prod_{c \in \mathcal{C}(S,u)} \frac{1}{1 - \rho_c(\mathbf{p})} \cdot \left(|p_u| + |p_u| \sum_{c \in \mathcal{C}(S,u)} \frac{\beta_c}{1 - \rho_c} \right) \\ &= \rho_{S,u}(\mathbf{p}) \cdot \left(1 + \sum_{c \in \mathcal{C}(S,u)} \frac{\beta_c}{1 - \rho_c} \right). \quad \square \end{aligned}$$

We will now prove several additional properties of the error sensitivity parameter.

Lemma 3.4. Fix a parameter vector $\mathbf{p} \in \mathcal{S}$. Let t_0 be such that for $0 \leq t \leq t_0$, $(1+t)\mathbf{p}$ is also in \mathcal{S} . Define $\beta_{S,u}(\mathbf{p}, t) = \frac{d}{dt} \rho_{S,u}((1+t)\mathbf{p})$. (Note that Definition 3.2 is consistent with $\beta_{S,u}(\mathbf{p}) = \beta_{S,u}(\mathbf{p}, 0)$.) Then, for all nodes (S,u) in the computation tree, $\beta_{S,u}(\mathbf{p}, t)$ is a non-negative,

non-decreasing function of t for $t \in [0, t_0]$. Thus, the map $t \mapsto \rho_{S,u}((1+t)\mathbf{p})$ is non-decreasing and convex over the same domain.

Proof. We induct on $|S|$. The base case is $S = \{u\}$, so $\rho_{\{u\},u}((1+t)\mathbf{p}) = (1+t)|p_u|$. We therefore have

$$\beta_{\{u\},u}(\mathbf{p}, t) = \frac{d}{dt} \rho_{\{u\},u}((1+t)\mathbf{p}) = |p_u|,$$

which is a constant (and hence non-decreasing), non-negative function of t .

For the inductive case, we use a recursive formula for $\beta_{S,u}(\mathbf{p}, t)$ as in the proof of Claim 3.3. We have

$$\begin{aligned} \beta_{S,u}(\mathbf{p}, t) &= |p_u| \cdot \prod_{(S',u') \in \mathcal{C}(S,u)} \frac{1}{1 - \rho_{S',u'}((1+t)\mathbf{p})} \\ &\quad + \rho_{S,u}((1+t)\mathbf{p}) \cdot \sum_{(S',u') \in \mathcal{C}(S,u)} \frac{\beta_{S',u'}(\mathbf{p}, t)}{1 - \rho_{S',u'}((1+t)\mathbf{p})}. \end{aligned}$$

By the induction hypothesis, $\beta_{S',u'}(\mathbf{p}, t) \geq 0$ for each $|S'| < |S|$, $u' \in S'$. Since $(1+t)\mathbf{p} \in \mathcal{S}$, Lemma 2.8 implies $0 \leq \rho_{S',u'}((1+t)\mathbf{p}) < 1$. Therefore $\beta_{S,u}(\mathbf{p}, t) \geq 0$ as well. Moreover, the inductive hypothesis implies that both $\rho_{S',u'}((1+t)\mathbf{p})$ and $\beta_{S',u'}(\mathbf{p}, t)$ are non-decreasing in t . Since the whole expression is monotone in $\rho_{S',u'}$ and $\beta_{S',u'}(\mathbf{p}, t)$, the left-hand side $\beta_{S,u}(\mathbf{p}, t)$ is also a non-decreasing function of t . \square

We can now prove the following relations between the $\beta_{S,u}$ and the $\rho_{S,u}$.

Lemma 3.5. Let $\mathbf{p} \in \mathbb{C}^V$ and $\alpha > 0$ satisfy $(1+\alpha)\mathbf{p} \in \mathcal{S}$. We then have the following inequalities for all nodes (S,u) in the computation tree. (We use the shorthand notation $\rho_{S,u} = \rho_{S,u}(\mathbf{p})$ and $\beta_{S,u} = \beta_{S,u}(\mathbf{p})$.)

1. $\beta_{S,u} < (1 - \rho_{S,u})/\alpha$.
2. $\rho_{S,u} \leq \beta_{S,u} \leq (1 + d_u/\alpha) \cdot \rho_{S,u}$, where d_u is the degree of the vertex u in G .
3. $\rho_{S,u} \leq \frac{1}{1+\alpha} \rho_{S,u}((1+\alpha)\mathbf{p}) < \frac{1}{1+\alpha}$.

Proof. Since $(1+\alpha)\mathbf{p} \in \mathcal{S}$, Lemma 2.8 implies $\rho_{S,u}((1+\alpha)\mathbf{p}) < 1$. Further, from Lemma 3.4, we know that $t \mapsto \rho_{S,u}((1+t)\mathbf{p})$ is convex for $t \in [0, \alpha]$. Item 1 of the lemma then follows from the inequalities

$$(3.3) \quad 1 > \rho_{S,u}((1+\alpha)\mathbf{p}) \geq \rho_{S,u} + \alpha \left. \frac{d\rho_{S,u}}{dt} \right|_{t=0} \geq \rho_{S,u} + \alpha \beta_{S,u}.$$

Item 2 follows from Claim 3.3, using $\beta_{S,u} \geq 0$, $|\mathcal{C}(S,u)| \leq d_u$, and $\beta_{S,u} < (1 - \rho_{S,u})/\alpha$ from item 1.

To prove the first inequality in item 3, we again use eq. (3.3), and substitute $\beta_{S,u} \geq \rho_{S,u}$ from item 2. The second inequality follows from the fact that $\rho_{S,u}((1+\alpha)\mathbf{p}) < 1$ as mentioned above. \square

Finally we relate the quantities $\rho_{S,u}$ to the quantities $r_{S,u}$ that we actually want to approximate.

Claim 3.6. Let \mathbf{p} lie in the complex Shearer region. For any node (S, u) in the computation tree, we have

$$|r_{S,u}(\mathbf{p})| \leq \rho_{S,u}(\mathbf{p}) \text{ and } |R_{S,u}(\mathbf{p})| \leq \rho_{S,u}(\mathbf{p}).$$

Proof. For both $r_{S,u}$ and $R_{S,u}$, the proof is by induction on $|S|$. The base case for $r_{S,u}$ is when S is a singleton, in which case we have $|r_{\{u\},u}(\mathbf{p})| = |p_u| = \rho_{\{u\},u}(\mathbf{p})$. For $R_{S,u}$, the base case is when (S, u) is at depth ℓ in the computation tree (where ℓ is as in the input to Algorithm 1), in which case one has $R_{S,u}(\mathbf{p}) = 0 \leq \rho_{S,u}(\mathbf{p})$. For the inductive case, we use the recursion for $r_{S,u}(\mathbf{p})$ to obtain

$$\begin{aligned} |r_{S,u}(\mathbf{p})| &= \left| p_u \prod_{c \in \mathcal{C}(S,u)} \frac{1}{1 - r_c(\mathbf{p})} \right| \\ &\leq |p_u| \prod_{c \in \mathcal{C}(S,u)} \frac{1}{1 - |r_c(\mathbf{p})|} \\ &\leq |p_u| \prod_{c \in \mathcal{C}(S,u)} \frac{1}{1 - \rho_c(\mathbf{p})} = \rho_{S,u}. \end{aligned}$$

Here, the first inequality follows from Fact C.1 since the induction hypothesis implies that $\mathbf{p} \in \mathcal{S}$ $|r_c(\mathbf{p})| \leq \rho_c(\mathbf{p})$, while that fact that $\mathbf{p} \in \mathcal{S}$ implies that $\rho_c(\mathbf{p}) < 1$ (e.g., from item 3 of Lemma 3.5). The second inequality follows directly from the induction hypothesis. The inductive step for $R_{S,u}$ is identical. \square

3.2 Correlation decay with complex activities

We now use the error sensitivity parameters to establish the correlation decay results needed for our FPTAS.

Let $G = (V, E)$ be a graph on n vertices, and let $\mathbf{p} \in \mathbb{C}^V$ be such that $(1 + \alpha)^2 \mathbf{p} \in \mathcal{S}$ (\mathbf{p} is in the Shearer region with slack $\simeq 2\alpha$). The root of the recursion is a pair (A, a) where $A \subseteq V$ and $a \in A$. Let $\ell \geq 0$ be arbitrary. Recall that Algorithm 1 recursively computes an estimate $R_{S,u}$ of $r_{S,u}$, where for every pair (S, u) encountered, we have $R_{S,u} = 0$ if (S, u) is at depth ℓ in the computation tree, and $R_{S,u} = p_u \cdot \prod_{c \in \mathcal{C}(S,u)} (1 - R_c)^{-1}$ otherwise. The depth δ of a node is defined as its distance from the root (A, a) in the recursive tree: the root has $\delta(A, a) = 0$, its children have $\delta(S, u) = 1$, etc.

Intuition. Claim 2.1 implies that it is sufficient to get good approximations for the occupation ratios in order to obtain an FPTAS. Suppose now that we were to expand the computation tree for computing a particular occupation ratio up to depth ℓ as described above, and were then able to show that at every node (S, u) in this tree, the approximation error $|r_{S,u} - R_{S,u}|$ is smaller than the maximum approximation error at the node's children by a factor $c < 1$. It would then follow that the approximation error at the

root node is $O(c^\ell)$, and hence that it is sufficient to take $\ell = O(\log n)$ in order to obtain an inverse polynomial approximation of the occupation ratio (which in turn can be shown to be sufficient for obtaining an FPTAS for Z). However, since degrees and the activity parameters might vary throughout the tree, the errors $|r_{S,u} - R_{S,u}|$ do not decay uniformly in this fashion at each node of the tree; they might even increase locally. (Examples are not difficult to construct.) Instead, we aim to use the error sensitivity parameter $\beta_{S,u}$ as a yardstick against which the approximation error $|r_{S,u} - R_{S,u}|$ at node (S, u) ought to be compared.

As we mentioned earlier, $\beta_{S,u}$ is a natural error sensitivity parameter because it satisfies a recurrence (Claim 3.3) similar to the recurrence for error propagation (Claim 3.1). In an earlier version of this paper, we used “normalized errors” roughly of the form $|r_{S,u} - R_{S,u}|/\beta_{S,u}$, and showed that they decay by a factor of $1 - \Theta(\alpha)$ at each level of the tree. Here we present an improved analysis which leads to a decay factor of $1 - \Theta(\sqrt{\alpha})$, which is in fact tight (in particular on the infinite d -regular tree, see Appendix F).

The improvement comes from analyzing in conjunction the behavior of $r_{S,u}$ at two different probability vectors: \mathbf{p} and the vector $(1 + \alpha)\mathbf{p}$ of slightly larger probabilities (which nonetheless still has a slack of α). We denote $\beta'_{S,u} = \left. \frac{d\rho_{S,u}((1+t)\mathbf{p})}{dt} \right|_{t=\alpha}$. Instead of $\beta_{S,u}$, we compare the errors to the quantity $\sqrt{\beta_{S,u}\beta'_{S,u}}$, i.e., we normalize the error at node (S, u) as $|r_{S,u} - R_{S,u}|/\sqrt{\beta_{S,u}\beta'_{S,u}}$.

The reason for this choice of the normalization is as follows. It can be shown that the case where the earlier version of the normalized error decays by a factor of only $1 - O(\alpha)$ corresponds to the situation where $\rho_{S,u}((1 + \alpha)\mathbf{p}) \simeq (1 + \alpha)\rho_{S,u}(\mathbf{p})$. But in that case, an argument based on Lemmas 3.4 and 3.5 implies that $\beta_{S,u} - \rho_{S,u}$ must be quite small. The new normalization of the error allows us to exploit this phenomenon: we can now show, roughly speaking, that the smaller of these two factors, i.e., $\rho_{S,u}(\mathbf{p})/\rho_{S,u}((1 + \alpha)\mathbf{p})$ (which was the factor obtained in the analysis of the earlier version) on the one hand, and $\beta_{S,u} - \rho_{S,u}$ on the other, can be taken to be the decay factor for the new normalized error. We then show that at least one of these two factors is as small as $1 - \Omega(\sqrt{\alpha})$. At a technical level, proving this requires a careful comparison of the recurrences for the propagation of unnormalized errors (Claim 3.1) with the recurrence for the $\beta_{S,u}$ (Claim 3.3), exploiting in particular the extra additive term of 1 in the latter recurrence. We now make this intuition precise in the following theorem. Recall that \mathbf{p} is assumed to be such that $(1 + \alpha)^2 \mathbf{p} \in \mathcal{S}$.

Theorem 3.7. For notational simplicity, let $\rho_{S,v} = \rho_{S,v}(\mathbf{p})$ and $\rho'_{S,v} = \rho_{S,v}((1 + \alpha)\mathbf{p})$. Similarly, let $\beta_{S,v} = \beta_{S,v}(\mathbf{p})$ and $\beta'_{S,v} = \beta_{S,v}((1 + \alpha)\mathbf{p})$. For a node (S, u) in a computation

tree of depth ℓ ,

$$|r_{S,u} - R_{S,u}| \leq \sqrt{\beta_{S,u} \beta'_{S,u}} (1 + \sqrt{\alpha})^{-(\ell - \delta(S,u))/2}.$$

Proof. The proof is by induction on $\ell - \delta(S,u)$. The base case is $\delta(S,u) = \ell$ and $R_{S,u} = 0$; we want to prove $|r_{S,u}| \leq \sqrt{\beta_{S,u} \beta'_{S,u}}$. We have $|r_{S,u}| \leq \rho_{S,u}$ (Claim 3.6). The base case follows since $\rho_{S,u} \leq \beta_{S,u}$ and $\rho_{S,u} \leq \rho'_{S,u} \leq \beta'_{S,u}$, by Lemma 3.4, and item 2 of Lemma 3.5.

For the inductive step, we apply the recursive formula from Claim 3.1:

$$|r_{S,u} - R_{S,u}| \leq \rho_{S,u} \sum_{c \in \mathcal{C}(S,u)} \frac{|r_c - R_c|}{1 - \rho_c}.$$

By definition, $\delta(c) = \delta(S,u) + 1$ for all $c \in \mathcal{C}(S,u)$. By the induction hypothesis, we therefore have

$$\begin{aligned} |r_{S,u} - R_{S,u}| &\leq \rho_{S,u} \sum_{c \in \mathcal{C}(S,u)} \frac{\sqrt{\beta_c \beta'_c}}{1 - \rho_c} (1 + \sqrt{\alpha})^{-(\ell - \delta(S,u) - 1)/2} \\ &\leq \frac{\rho_{S,u} \sum_{c \in \mathcal{C}(S,u)} \sqrt{\frac{\beta_c}{1 - \rho_c}} \cdot \sqrt{\frac{\beta'_c}{1 - \rho'_c}}}{(1 + \sqrt{\alpha})^{(\ell - \delta(S,u) - 1)/2}}, \\ &\leq \frac{\rho_{S,u} \sqrt{\sum_{c \in \mathcal{C}(S,u)} \frac{\beta_c}{1 - \rho_c}} \cdot \sqrt{\sum_{c \in \mathcal{C}(S,u)} \frac{\beta'_c}{1 - \rho'_c}}}{(1 + \sqrt{\alpha})^{(\ell - \delta(S,u) - 1)/2}} \\ (3.4) \quad &= \frac{\sqrt{\frac{\rho_{S,u}}{\rho'_{S,u}}} \sqrt{\beta_{S,u} - \rho_{S,u}} \cdot \sqrt{\beta'_{S,u} - \rho'_{S,u}}}{(1 + \sqrt{\alpha})^{(\ell - \delta(S,u) - 1)/2}} \end{aligned}$$

where the second inequality uses $\rho_c \leq \rho'_c$ (which follows from item 3 of Lemma 3.5), the third is the Cauchy-Schwarz inequality, and the last equality uses the recursion for $\beta_{S,u}$ as developed in Claim 3.3.³ Note that it also follows from items 2 and 3 of Lemma 3.5 that

$$(3.5) \quad 0 \leq \rho_{S,u} \leq \rho'_{S,u} \text{ and } \beta_{S,u} - \rho_{S,u}, \beta'_{S,u} - \rho'_{S,u} \geq 0.$$

We now divide the rest of the analysis into two cases.

Case 1: $\beta_{S,u} \leq \rho_{S,u}/\sqrt{\alpha}$. In this case, we have $0 \leq \beta_{S,u} - \rho_{S,u} \leq \beta_{S,u}(1 - \sqrt{\alpha}) \leq \beta_{S,u}/(1 + \sqrt{\alpha})$. Substituting this into eq. (3.4), and estimating the rest of the factors using (3.5), we get

$$\begin{aligned} |r_{S,u} - R_{S,u}| &\leq \frac{\sqrt{\frac{\rho_{S,u}}{\rho'_{S,u}}} \sqrt{\beta_{S,u} - \rho_{S,u}} \cdot \sqrt{\beta'_{S,u} - \rho'_{S,u}}}{(1 + \sqrt{\alpha})^{(\ell - \delta(S,u) - 1)/2}} \\ &\leq \frac{\sqrt{\beta_{S,u} \beta'_{S,u}}}{(1 + \sqrt{\alpha})^{(\ell - \delta(S,u) - 1)/2}} \cdot \frac{1}{\sqrt{1 + \sqrt{\alpha}}}, \end{aligned}$$

which completes the inductive step in this case.

³We implicitly assume here and later in this proof that $\rho'_{S,u}$ is strictly positive. For, if $\rho'_{S,u}$ were 0, then by item 3 of Lemma 3.5, $\rho_{S,u}$ and $R_{S,u}$ will also be 0 and the inductive hypothesis will be trivially true.

Case 2: $\beta_{S,u} > \rho_{S,u}/\sqrt{\alpha}$. We claim that in this case, $\rho_{S,u} \leq \rho'_{S,u}/(1 + \sqrt{\alpha})$. Indeed, using the same argument as in the proof of item 1 of Claim 3.3, we have

$$(3.6) \quad \begin{aligned} \rho'_{S,u} &:= \rho_{S,u}((1 + \alpha)\mathbf{p}) \geq \rho_{S,u} + \alpha \frac{d\rho_{S,u}}{dt} \Big|_{t=0} \\ &\geq \rho_{S,u} + \alpha \beta_{S,u} > (1 + \sqrt{\alpha})\rho_{S,u}. \end{aligned}$$

Substituting this into eq. (3.4), we obtain

$$\begin{aligned} |r_{S,u} - R_{S,u}| &\leq \frac{\sqrt{\frac{\rho_{S,u}}{\rho'_{S,u}}} \sqrt{\beta_{S,u} - \rho_{S,u}} \cdot \sqrt{\beta'_{S,u} - \rho'_{S,u}}}{(1 + \sqrt{\alpha})^{(\ell - \delta(S,u) - 1)/2}} \\ &\leq \frac{1}{\sqrt{1 + \sqrt{\alpha}}} \cdot \frac{\sqrt{\beta_{S,u} \beta'_{S,u}}}{(1 + \sqrt{\alpha})^{(\ell - \delta(S,u) - 1)/2}}, \end{aligned}$$

which establishes the induction step in this case as well. \square

Corollary 3.8. Given a graph $G = (V, E)$, let \mathbf{p} be a complex parameter vector such that $(1 + \alpha)^2 \mathbf{p} \in \mathcal{S}$. Let (A, a) be the root of the recursive computation of Theorem 3.7, where a is a vertex of degree d_a in G . Then, we have

$$|r_{A,a} - R_{A,a}| \leq \frac{1 + d_a/\alpha}{(1 + \sqrt{\alpha})^{\ell/2}}.$$

Proof. Recall from Lemma 2.8 that $\rho_{A,a} \in [0, 1)$ when $\mathbf{p} \in \mathcal{S}$. Then item 2 of Lemma 3.5 implies that

$$\beta_{A,a}((1 + \alpha)\mathbf{p}) \leq (1 + d_a/\alpha) \cdot r_{A,a}((1 + \alpha)\mathbf{p}) \leq 1 + d_a/\alpha.$$

We can apply Lemma 3.5 to $\beta_{A,a}((1 + \alpha)\mathbf{p})$ (as opposed to $\beta_{A,a}(\mathbf{p})$) because of the assumption that $(1 + \alpha)^2 \mathbf{p} \in \mathcal{S}$. The claim now follows from Theorem 3.7 since (A, a) is at depth 0 in the computation tree. \square

We can now prove that our algorithm indeed provides an FPTAS for the quantity $\check{q}_V(\mathbf{p})$ (for bounded degree graphs and constant slack). We remark that this also proves Theorem 1.1.

Theorem 3.9 (FPTAS for \check{q}). Given $\alpha, \varepsilon \in (0, 1]$, a graph $G = (V, E)$ on n vertices with maximum degree d , and a parameter vector \mathbf{p} such that $(1 + \alpha)^2 \mathbf{p} \in \mathcal{S}$, a $(1 + \varepsilon)$ -approximation to $\check{q}_V(\mathbf{p})$ can be computed in time $(\frac{n}{\varepsilon \alpha})^{O(\log(d)/\sqrt{\alpha})}$.

Proof. Order the vertices of G arbitrarily as v_1, v_2, \dots, v_n . Recall that

$$\check{q} := \check{q}_V = \prod_{i=1}^n (1 - r_{S_i, v_i}),$$

where $S_i := \{v_i, v_{i+1}, \dots, v_n\}$. Let us denote by capital letters the estimates computed by Algorithm 1. R_{S_i, v_i} is

computed using ℓ levels of the recurrence in Theorem 3.7, where

$$\ell = \left\lceil 2 \log_{(1+\sqrt{\alpha})} \left(\frac{2(1+\alpha)(1+d/\alpha)n}{\varepsilon\alpha} \right) \right\rceil.$$

We have

$$\ell = \left\lceil \frac{2}{\log(1+\sqrt{\alpha})} \log \left(\frac{2(1+\alpha)(1+d/\alpha)n}{\varepsilon\alpha} \right) \right\rceil \leq O \left(\frac{1}{\sqrt{\alpha}} \log \left(\frac{n}{\varepsilon\alpha} \right) \right).$$

The number of nodes of the computation tree explored in the computation of each R_{S_i, v_i} is $O(d^\ell)$ since the graph is assumed to be of degree at most d . This proves the running time bound.

The algorithm outputs the quantity

$$\check{Q} := \prod_{i=1}^n (1 - R_{S_i, v_i}).$$

We now prove that this is indeed a $(1+\varepsilon)$ -approximation for $\check{q}_V(\mathbf{p})$. For ease of notation, we define $\xi_i := 1 - r_{S_i, v_i}$ and $\Xi_i = 1 - R_{S_i, v_i}$. From Corollary 3.8 we have, for each i ,

$$(3.7) \quad \left| \frac{\Xi_i}{\xi_i} - 1 \right| = \frac{|r_{S_i, v_i} - R_{S_i, v_i}|}{|1 - r_{S_i, v_i}|} \leq \frac{1+\alpha}{\alpha} \cdot \frac{1+d/\alpha}{(1+\sqrt{\alpha})^{\ell/2}}.$$

Here, the last inequality uses Claim 3.6 (which implies that $|r_{S_i, v_i}| \leq \rho_{S_i, v_i}$) and item 3 from Lemma 3.5 (which implies that when $\mathbf{p} \in \mathcal{S}$, $\rho_{S_i, v_i} \leq \frac{1}{1+\alpha} < 1$). Together, with Fact C.1, these two inequalities imply that $\frac{1}{|1 - r_{S_i, v_i}|} \leq \frac{1+\alpha}{\alpha}$.

Since $\ell = \left\lceil 2 \log_{(1+\sqrt{\alpha})} \left(\frac{2(1+\alpha)(1+d/\alpha)n}{\varepsilon\alpha} \right) \right\rceil$, we therefore have $\left| \frac{\Xi_i}{\xi_i} - 1 \right| \leq \frac{\varepsilon}{2n}$, for all i . Combining this with Fact C.2, and recalling that $\check{q} = \prod_{i=1}^n \xi_i$ and that $\check{Q} = \prod_{i=1}^n \Xi_i$, we obtain $|\check{Q} - \check{q}| \leq \varepsilon |\check{q}|$, which proves the theorem. \square

4 Application for the Lovász Local Lemma

4.1 Proof of Shearer's lemma by rounding variables

Let us recall Shearer's formulation [35] of the Lovász Local Lemma, as stated in Appendix A. For any distribution μ and events $\mathcal{E}_1, \dots, \mathcal{E}_n$ with dependency graph G , and any $\mathbf{p} \in \mathcal{S}_G$ for which $\mu(\mathcal{E}_i) \leq p_i$, then $\mu(\bigcap_{i=1}^n \overline{\mathcal{E}_i}) > 0$, and hence $\bigcap_{i=1}^n \overline{\mathcal{E}_i} \neq \emptyset$.

In this section, we give a new proof of Shearer's lemma in the so-called "variable model", in which it is assumed that the events are determined by underlying independent variables. The variable model is assumed in most algorithmic formulations of the Lovász Local Lemma [29].

For the purposes of this section, it will be slightly more convenient to use the definition of the Shearer region from Definition 2.4:

$$(4.1) \quad \mathcal{S} = \{ \mathbf{p} \in \mathbb{R}^V : \check{q}_S(\mathbf{p}) > 0 \ \forall S \subseteq V \}.$$

Preliminaries. We will use the variable model, as described in Appendix A.1. For simplicity we restrict to $\{0, 1\}$ -valued variables, although similar arguments work for variables with arbitrary finite domains. Given any vector $\mathbf{z} \in [0, 1]^m$, let $\mu_{\mathbf{z}}$ now be the product distribution on $\Omega = \{0, 1\}^m$ with expectation \mathbf{z} . We assume that event \mathcal{E}_i depends only on the coordinates $A_i \subseteq [m]$. The dependency graph G on vertex set $V = [n]$ has an edge between i and j if $A_i \cap A_j \neq \emptyset$.

Multilinearity. Let us now define $\mathbf{p} = \mathbf{p}(\mathbf{z}) \in [0, 1]^n$ by $p_i = \mu_{\mathbf{z}}(\mathcal{E}_i)$. We first observe that each p_i is a multilinear polynomial in \mathbf{z} :

$$(4.2) \quad p_i = p_i(\mathbf{z}) = \sum_{S \in \Pi_{A_i}(\mathcal{E}_i)} \prod_{j \in S} z_j \prod_{j \in A_i \setminus S} (1 - z_j),$$

where Π_{A_i} denotes projection to the coordinates in A_i .

The key observation is that $\check{q}_U(\mathbf{p}(\mathbf{z}))$ is also a multilinear polynomial in \mathbf{z} , for any $U \subseteq V$. To see this, note that the events depending on variable z_j form a clique in G , whereas each summand $\pm \prod_{i \in I} p_i$ in the definition of \check{q}_U involves an independent set I in G . Since a clique and an independent set intersect in at most one vertex, each summand $\pm \prod_{i \in I} p_i$ involves at most one p_i that depends on z_j . So $\pm \prod_{i \in I} p_i$ is multilinear in \mathbf{z} , and the same is true of $\check{q}_U(\mathbf{p}(\mathbf{z}))$.

Proof of Shearer's Lemma. In the variable model, the hypothesis of Shearer's lemma is that there exists $\mathbf{z} \in [0, 1]^m$ such that under distribution $\mu_{\mathbf{z}}$ we have $\mathbf{p}(\mathbf{z}) \in \mathcal{S}$. The conclusion $\bigcap_{i=1}^n \overline{\mathcal{E}_i} \neq \emptyset$ is equivalent to the existence of an assignment $\mathbf{z} \in \{0, 1\}^m$ to the variables with $\mathbf{p}(\mathbf{z}) = \mathbf{0}$.

We prove that $\bigcap_{i=1}^n \overline{\mathcal{E}_i} \neq \emptyset$ by the following argument. Given an initial vector $\mathbf{z} \in [0, 1]^m$ with $\mathbf{p}(\mathbf{z}) \in \mathcal{S}$, we first round z_1 to 0 or 1 while maintaining the property that $\mathbf{p}(\mathbf{z}) \in \mathcal{S}$. Then we repeat with z_2, \dots, z_m , resulting in a final vector $\mathbf{z} \in \{0, 1\}^m$ with $\mathbf{p} = \mathbf{p}(\mathbf{z}) \in \mathcal{S}$. As the distribution $\mu_{\mathbf{z}}$ is now deterministic, we must have $\mathbf{p} \in \{0, 1\}^n$. In fact $\mathbf{p} = \mathbf{0}$, for if $p_j = 1$ then $\check{q}_{\{j\}} = 1 - p_j = 0$, contradicting that $\mathbf{p} \in \mathcal{S}$. Thus none of the events $\mathcal{E}_1, \dots, \mathcal{E}_n$ occur under the assignment \mathbf{z} , so $\mathbf{z} \in \bigcap_{i=1}^n \overline{\mathcal{E}_i}$.

The crux is deciding how to round z_i . We will increase z_i to 1 if $\frac{\partial \check{q}_V}{\partial z_i}(\mathbf{p}(\mathbf{z})) \geq 0$, and otherwise decrease z_i to 0. This decision ensures that \check{q}_V does not decrease during this rounding step, since \check{q}_V is multilinear in \mathbf{z} . Note that the condition $\check{q}_V(\mathbf{p}) > 0$ alone does not imply that $\mathbf{p} \in \mathcal{S}$; referring to Equation (4.1), we must also ensure that $\check{q}_S(\mathbf{p}) > 0$ for all $S \subseteq V$. Fortunately Lemma 2.7 implies that $\check{q}_V(\mathbf{p}) = \min_{S \subseteq V} \check{q}_S(\mathbf{p})$ whenever $\mathbf{p} \in \mathcal{S}$. So,

thinking of *continuously* modifying z_i , if any $\check{q}_S(\mathbf{p})$ were to become non-positive then $\check{q}_V(\mathbf{p})$ should be the first to do so. Since the rounding is a continuous process ensuring that $\check{q}_V(\mathbf{p}) > 0$, this is actually sufficient to imply that $\mathbf{p} \in \mathcal{S}$. A formal version of this argument appears in Appendix D.1.

Remarks. Since this rounding argument for the LLL in Shearer’s region is very simple, one might be tempted to try a similar rounding in the region \mathcal{L} employed in the original form of the Lovász Local Lemma (see Appendix A). It turns out that \mathcal{L} does not support such roundings: there is a probability space in the variable model with $\mathbf{p} \in \mathcal{L}$, such that rounding some variable to 0 or 1 will *both* lead to $\mathbf{p} \notin \mathcal{L}$. An example is shown in Appendix D.2. So the fact that our rounding argument works is a special property of the Shearer region.

This proof of Shearer’s lemma directly suggests a potential algorithm: compute the sign $\frac{\partial \check{q}_V}{\partial z_i}(\mathbf{p}(\mathbf{z}))$ in order to perform the rounding. We design such an algorithm in the next section.

4.2 An algorithmic LLL by polynomial evaluation

We now explain how the algorithm from the previous subsection can be made to run in subexponential time, using our FPTAS COMPUTEINDEPPOLY, assuming that the initial probabilities have slack. We also assume that the probabilities $p(\mathbf{z})$ of the events can be efficiently computed given the probability distribution \mathbf{z} on the underlying variables: this is the case in standard applications of the variable model LLL such as k -CNF-SAT. The notation used in the theorem is as in the previous subsection.

Theorem 4.1. Assume that the initial distribution \mathbf{z} has slack $\alpha \in (0, 1]$, i.e., $(1 + \alpha) \cdot \mathbf{p}(\mathbf{z}) \in \mathcal{S}$, and that $\mathbf{p}(\mathbf{z})$ can be computed from \mathbf{z} in time polynomial in m . Then there is a deterministic algorithm with runtime $(nm/\alpha)^{O(\log(d)\sqrt{m/\alpha})}$ that can construct a point in $\bigcap_{i=1}^n \overline{\mathcal{E}}_i$. Here, d is the degree of the dependency graph.

Main ideas. Recall that the algorithm examines the sign of $\frac{\partial \check{q}_{[n]}}{\partial z_i}(\mathbf{p}(\mathbf{z}))$ in order to decide whether to round z_i up or down. Since $\check{q}_{[n]}$ is multilinear in z_i , we can estimate $\frac{\partial \check{q}_{[n]}}{\partial z_i}(\mathbf{p}(\mathbf{z}))$ by using the FPTAS to compute $\check{q}_{[n]}$ at two points nearby $\mathbf{p}(\mathbf{z})$.

Recall that the FPTAS is only efficient so long as there is sufficiently large slack. So the main challenge in the rounding is ensuring that the points $\mathbf{p}(\mathbf{z})$ constructed during the algorithm not only remain in \mathcal{S} but also have slack. In each of the m iterations, we might step a bit towards the Shearer boundary, but we ensure that in one step, the slack cannot decrease by more than $\frac{\alpha}{2m}$. Since the initial slack is at least α , it can then be insured that

all points constructed during the m iterations have slack at least $\frac{\alpha}{m}$.

Detailed discussion. The input to the algorithm is \mathbf{z}^0 satisfying $(1 + \alpha) \cdot \mathbf{p}(\mathbf{z}^0) \in \mathcal{S}$. As a preprocessing step, we will first eliminate any coordinates of \mathbf{z} that are nearly integral: if $z_i \leq \alpha/4$ we set z_i to 0, and if $z_i \geq 1 - \alpha/4$ we set z_i to 1. In doing so, $p_j(\mathbf{z})$ can increase by at most a factor $(1 - \frac{\alpha}{4})^{-1}$, because p_j is a probability and is multilinear in \mathbf{z} . So the resulting point $\mathbf{p}(\mathbf{z})$ still has slack at least $\frac{\alpha}{2}$.

As in the non-constructive version above, the algorithm has m iterations, in which the i^{th} iteration rounds z_i to either 0 or 1. Define $s_i = 1 + \frac{\alpha(m-i)}{2m}$. We maintain the following invariant:

(†) At the start of iteration i ,
the point $\mathbf{p}(\mathbf{z})$ has slack at least $s_{i-1} - 1$.

We then proceed to estimate $\frac{\partial \check{q}_{[n]}}{\partial z_i}$ at the point $s_i \cdot \mathbf{p}(\mathbf{z})$ which, due to the definition of s_i , still has slack $\frac{\alpha}{4m}$. Note that

$$(4.3) \quad \frac{\partial \check{q}_{[n]}}{\partial z_i}(s_i \cdot \mathbf{p}(\mathbf{z})) = \frac{1}{\delta} \left(\check{q}_{[n]}(s_i \cdot \mathbf{p}(\mathbf{z})) - \check{q}_{[n]}(s_i \cdot \mathbf{p}(\mathbf{z} - \delta \mathbf{e}_i)) \right),$$

by multilinearity. Our algorithm will choose an appropriate δ , then use the FPTAS to estimate the two terms on the right-hand side with sufficiently small error.

First we must check that $s_i \cdot \mathbf{p}(\mathbf{z} - \delta \mathbf{e}_i)$ is still in the Shearer region, and with sufficient slack. Set $\delta = \frac{\alpha^2}{36m}$. Since $z_i \geq \alpha/4$, we certainly have $\mathbf{p}(\mathbf{z} - \delta \mathbf{e}_i) \geq 0$. Next we will prove that

$$(4.4) \quad s_i \cdot \mathbf{p}(\mathbf{z} - \delta \mathbf{e}_i) \leq (1 + \frac{\alpha}{9m}) s_i \cdot \mathbf{p}(\mathbf{z})$$

coordinate-wise. Since $s_i \cdot \mathbf{p}(\mathbf{z})$ has slack $\frac{\alpha}{4m}$, the inequality (4.4) will then imply that $s_i \cdot \mathbf{p}(\mathbf{z} - \delta \mathbf{e}_i)$ has slack $\frac{\alpha}{8m}$.

Let us consider the j th coordinate in (4.4). Fixing all coordinates of \mathbf{z} other than z_i , we may write p_j as the linear function $\mu z_i + \nu$, where $\mu + \nu \geq 0$ since p_j is a probability. We may assume that $\mu \leq 0$, otherwise the inequality (4.4) is trivial. Then, using $\mu \leq 0$ and recalling that $z_i \leq 1 - \frac{\alpha}{4}$, we have

$$\begin{aligned} \frac{\mu(z_i - \delta) + \nu}{\mu z_i + \nu} &= 1 - \frac{\mu \delta}{\mu z_i + \nu} \leq 1 - \frac{\mu \delta}{\mu(1 - \frac{\alpha}{4}) + \nu} \\ &\leq 1 + \frac{\delta}{\alpha/4} = 1 + \frac{\alpha}{9m}. \end{aligned}$$

This proves (4.4).

Now let us return to our estimation of (4.3). For simplicity let us rewrite (4.3) with the following shorthand notation

$$\underbrace{\frac{\partial \check{q}_{[n]}}{\partial z_i}(s_i \cdot \mathbf{p}(\mathbf{z}))}_a = \frac{1}{\delta} \left(\underbrace{\check{q}_{[n]}(s_i \cdot \mathbf{p}(\mathbf{z}))}_{q_0} - \underbrace{\check{q}_{[n]}(s_i \cdot \mathbf{p}(\mathbf{z} - \delta \mathbf{e}_i))}_{q_\delta} \right)$$

$$\iff a = \frac{q_0 - q_\delta}{\delta}.$$

We set $\varepsilon = \frac{\delta}{4}$, then use the FPTAS to compute quantities $Q_0 \in [1 - \varepsilon, 1] \cdot q_0$ and $Q_\delta \in [1 - \varepsilon, 1] \cdot q_\delta$, then compute

$$A = \frac{Q_0 - Q_\delta}{\delta}.$$

It then follows that

$$(4.5) \quad -\frac{\varepsilon}{\delta}q_\delta \leq a - A \leq \frac{\varepsilon}{\delta}q_0.$$

The algorithm's rounding proceeds as follows. Let $\mathbf{z}' = \mathbf{z}$. If $A \geq 0$ then round z'_i to 1, otherwise round z'_i to 0. We claim that \mathbf{z}' satisfies $s_i \cdot \mathbf{p}(\mathbf{z}') \in \mathcal{S}$. This implies that \mathbf{z}' has slack at least $s_i - 1$, so that the invariant (\dagger) is satisfied at the start of the next (i.e., the $(i+1)$ th) iteration.

To prove this claim, we begin by observing that Lemma D.1 implies that it is sufficient to prove that while rounding the i th co-ordinate, $\check{q}_{[n]}(s_i \cdot \mathbf{p}(\mathbf{z}''))$ is strictly positive on all points \mathbf{z}'' lying on the straight line segment along which z_i is rounded. Let \mathbf{z}'' be a point on this line segment. Consider first the case $A < 0$. Using eq. (4.5), we have

$$\begin{aligned} \check{q}_{[n]}(s_i \cdot \mathbf{p}(\mathbf{z}'')) &= q_0 - (z_i - z''_i)a \geq q_0(1 - z_i \frac{\varepsilon}{\delta}) \\ &\geq q_0/2 > 0. \end{aligned}$$

Next consider the case $A \geq 0$. Then $0 \leq Q_0 - Q_\delta \leq q_0 - (1 - \varepsilon)q_\delta$, and hence $q_\delta \leq (1 + 2\varepsilon)q_0 \leq 2q_0$. Then, by eq. (4.5), we have $a \geq -\frac{\varepsilon}{\delta}q_\delta \geq -\frac{2\varepsilon}{\delta}q_0$. It follows that

$$\begin{aligned} \check{q}_{[n]}(s_i \cdot \mathbf{p}(\mathbf{z}'')) &= q_0 + (z''_i - z_i)a \geq q_0(1 - (1 - z_i)\frac{2\varepsilon}{\delta}) \\ &\geq q_0/2 > 0. \end{aligned}$$

This completes the claim that $s_i \cdot \mathbf{p}(\mathbf{z}') \in \mathcal{S}$.

Runtime. The FPTAS is invoked $O(m)$ times, each time with $\varepsilon = \frac{\delta}{4} = \frac{\alpha^2}{144m}$ and with slack at least $\alpha/8m$. The runtime is therefore

$$\begin{aligned} O(m) \cdot \left(\frac{n}{\varepsilon(\alpha/8m)} \right)^{O(\log(d)/\sqrt{\alpha/8m})} \\ = \left(\frac{nm}{\alpha} \right)^{O(\log(d)\sqrt{m/\alpha})}. \end{aligned}$$

5 Testing membership in Shearer's region

In this section we consider the following question:

Question. Given a graph G and activities p_v for all $v \in V$, is \mathbf{p} in the Shearer region of G ?

We recall that $\mathbf{p} \in \mathcal{S}_G$ if and only if $\check{q}_S(\mathbf{p}) > 0$ for all $S \subseteq [n]$, or equivalently if $\check{q}_V(\mathbf{p}) > 0$ everywhere on the line segment connecting $\mathbf{0}$ and \mathbf{p} . As we prove, it is #P-hard to answer this question exactly (see Appendix E).

On the other hand, in running time $2^{O(n)}$, we can trivially compute all Shearer's polynomials $\check{q}_S(\mathbf{p})$ and answer this question. Here we show that we can test membership approximately in subexponential time.

5.1 An algorithm to test membership in Shearer's region

Theorem 5.1. Given a graph G , $p_v \in (0, 1)$ for $v \in V$, and $0 < \alpha < 1$, there exists a deterministic algorithm which, in running time $(n/\alpha)^{O(\sqrt{n/\alpha} \log d)}$, decides whether $\mathbf{p} \in \mathcal{S}_G$ or $(1 + \alpha)\mathbf{p} \notin \mathcal{S}_G$.

Notation. It will be convenient to express some of the arguments in this section in terms of Shearer polynomials q defined as follows (see, e.g. [21]). For any subset S of vertices in a graph $G = (V, E)$,

$$(5.1) \quad q_S(\mathbf{p}) := \sum_{\substack{I \in \text{Ind}(G) \\ I \supseteq S}} (-1)^{|I \setminus S|} \prod_{i \in S} p_i.$$

Note that these polynomials are related to the \check{q} as follows:

$$q_S(\mathbf{p}) = \begin{cases} \left(\prod_{i \in S} p_i \right) \check{q}_{V \setminus \Gamma^+(S)} & \text{if } S \in \text{Ind}(G), \\ 0 & \text{otherwise.} \end{cases}$$

We want to harness our evaluation algorithm for $q_\emptyset(\mathbf{p}) = \check{q}_V(\mathbf{p})$. However, we need to proceed carefully, since evaluating $q_\emptyset(\mathbf{p})$ without knowing a lower bound on the slack might give the wrong answer. Hence we start from a point which is guaranteed to be in the Shearer region, and we maintain a lower bound on the slack at each point. We will use the following lemma.

Lemma 5.2. For a point $\mathbf{p} \in \mathcal{S}_G$, let $\gamma(\mathbf{p}) = 1 / \sum_{i=1}^n \frac{q_{\{i\}}(\mathbf{p})}{q_\emptyset(\mathbf{p})}$. Then the slack of \mathbf{p} is between $\gamma(\mathbf{p})$ and $n\gamma(\mathbf{p})$, or in other words: $(1 + \gamma(\mathbf{p}) - \varepsilon)\mathbf{p} \in \mathcal{S}_G$ for every $\varepsilon > 0$, and $(1 + n\gamma(\mathbf{p}))\mathbf{p} \notin \mathcal{S}_G$.

Proof. We know from [21] that $q_\emptyset((1+t)\mathbf{p})$ is a convex function of t , and

$$\begin{aligned} \frac{d}{dt} q_\emptyset((1+t)\mathbf{p}) \Big|_{t=0} &= \sum_{i=1}^n p_i \frac{\partial q_\emptyset}{\partial p_i}(\mathbf{p}) = - \sum_{i=1}^n p_i \check{q}_{V \setminus \Gamma^+(i)}(\mathbf{p}) \\ &= - \sum_{i=1}^n q_{\{i\}}(\mathbf{p}). \end{aligned}$$

Therefore, for $\lambda < \gamma(\mathbf{p}) = 1 / \sum_{i=1}^n \frac{q_{\{i\}}(\mathbf{p})}{q_\emptyset(\mathbf{p})}$, we have

$$\begin{aligned} \check{q}_V((1+\lambda)\mathbf{p}) &= q_\emptyset((1+\lambda)\mathbf{p}) \\ &\geq q_\emptyset(\mathbf{p}) + \lambda \frac{d}{dt} q_\emptyset((1+t)\mathbf{p}) \Big|_{t=0} \\ &= q_\emptyset(\mathbf{p}) - \lambda \sum_{i=1}^n q_{\{i\}}(\mathbf{p}) > 0. \end{aligned}$$

Along with Lemma D.1, this implies that $(1 + \gamma(\mathbf{p}) - \varepsilon)\mathbf{p} \in \mathcal{S}_G$ for every $\varepsilon > 0$, since $\mathbf{p} \in \mathcal{S}_G$ and $q_\emptyset((1 + \lambda)\mathbf{p}) > 0$ for every $\lambda \in [0, \gamma(\mathbf{p})]$.

On the other hand, we have $n\gamma(\mathbf{p}) \geq \frac{q_\emptyset(\mathbf{p})}{q_{\{i^*\}}(\mathbf{p})}$ where $q_{\{i^*\}}(\mathbf{p}) = \max_i q_{\{i\}}(\mathbf{p})$. Since $q_\emptyset(\mathbf{p})$ is linear in p_{i^*} , we obtain

$$q_\emptyset\left(\mathbf{p} + \frac{q_\emptyset(\mathbf{p})}{q_{\{i^*\}}(\mathbf{p})} p_{i^*} \mathbf{e}_{i^*}\right) = q_\emptyset(\mathbf{p}) + p_{i^*} \frac{q_\emptyset(\mathbf{p})}{q_{\{i^*\}}(\mathbf{p})} \frac{\partial q_\emptyset}{\partial p_{i^*}} = 0,$$

since $p_i \frac{\partial q_\emptyset}{\partial p_{i^*}} = -q_{\{i^*\}}(\mathbf{p})$. Thus, by monotonicity of the Shearer region, $(1 + n\gamma(\mathbf{p}))\mathbf{p} \notin \mathcal{S}_G$. \square

Now we can prove Theorem 5.1.

Proof. We maintain a point $\tilde{\mathbf{p}}$ such that $(1 + \frac{\alpha}{8n})\tilde{\mathbf{p}} \in \mathcal{S}_G$. We start with $\tilde{\mathbf{p}} = \frac{1}{2n} \cdot \mathbf{p}$ (where \mathbf{p} is the input vector) for which the statement is certainly true (since $(\frac{1}{n}, \dots, \frac{1}{n}) \in \mathcal{S}_G$ for any graph G). Note further that this initial point has slack $O(n)$, since if each p_v was less than $1/n$, $\mathbf{p} \in \mathcal{S}$ is trivially true.

Given $\tilde{\mathbf{p}}$, we compute $q_\emptyset(\tilde{\mathbf{p}})$ and $q_{\{i\}}(\tilde{\mathbf{p}})$ for all i within a relative error of $\varepsilon = 1/2$. More precisely, we obtain estimates within $[1, 3/2]$ times the correct value.

Thus we can also compute $\gamma(\tilde{\mathbf{p}}) = 1/\sum_{i=1}^n \frac{q_{\{i\}}(\tilde{\mathbf{p}})}{q_\emptyset(\tilde{\mathbf{p}})}$ within $[2/3, 3/2]$ times the correct value. Let us call this estimate $\tilde{\gamma}$. If $\tilde{\gamma} \leq \frac{\alpha}{2n}$ then we can conclude by Lemma 5.2 that $(1 + \alpha)\tilde{\mathbf{p}} \notin \mathcal{S}_G$.

If on the other hand $\tilde{\gamma} > \frac{\alpha}{2n}$, we know by Lemma 5.2 that $(1 + \frac{\alpha}{3n})\tilde{\mathbf{p}} \in \mathcal{S}_G$, and hence $(1 + \frac{\alpha}{6n})\tilde{\mathbf{p}}$ still has a slack of say $\frac{\alpha}{8n}$. In this case we replace $\tilde{\mathbf{p}}$ by $(1 + \frac{\alpha}{6n})\tilde{\mathbf{p}}$ and continue.

Let us analyze the running time. Whenever we evaluate $q_\emptyset(\tilde{\mathbf{p}}) = \check{q}_V(\tilde{\mathbf{p}})$ and $q_{\{i\}}(\tilde{\mathbf{p}}) = p_i \check{q}_{V \setminus \Gamma^+(i)}(\tilde{\mathbf{p}})$, we have a guaranteed slack of $\Omega(\alpha/n)$. By Theorem 3.9, we can evaluate these polynomials within a relative error of $\varepsilon = 1/2$ in running time $(\frac{2n}{\alpha})^{4\sqrt{n/\alpha} \log d} = (n/\alpha)^{O(\sqrt{n/\alpha} \log d)}$. Every time we iterate, the actual slack goes down by a factor of $1 - \Omega(1/n)$ (from Lemma 5.2). Since the multiplicative slack at the beginning is $O(n)$, and we stop when the slack becomes $O(\alpha/n)$, the number of iterations is $O(n \log(n/\alpha))$. Thus the total running time is $(n/\alpha)^{O(\sqrt{n/\alpha} \log d)}$. \square

6 Conclusions and open questions

The main open question left open by our work is whether the dependence on the slack in Theorem 1.1 can be improved from sub-exponential to polynomial. In all our applications of Theorem 1.1, we have to work with sub-constant values of the slack α , and it is the sub-exponential dependence on $1/\alpha$ that prevents us from giving polynomial time algorithms for these applications.

Question 1. Is there an algorithm to estimate $\check{q}_V(\mathbf{p})$ up to a $(1 + \varepsilon)$ -multiplicative factor in n -vertex graphs of maximum degree d , assuming that $(1 + \alpha)\mathbf{p} \in \mathcal{S}$, in running time $(\frac{n}{\alpha\varepsilon})^{O(\log d)}$?

In Appendix F, we provide some evidence to show that the correlation decay technique may not be capable of completely removing the sub-exponential dependence on $1/\alpha$. On the other hand, the result of Patel and Regts [31], to the best of our knowledge, has an even worse exponential dependence on $1/\alpha$, which, as discussed in Section 1.2, appear to be intrinsic to the techniques used there. A ray of hope is however offered by some recent progress in the positive activity setting, where Efthymiou, Hayes, Štefankovič, Vigoda and Yin [11] were able to obtain an FPRAS for the independence polynomial on graphs of large enough bounded degree and large enough girth, the exponent of whose running time has no dependence on the analog of the slack in the positive activity setting (the slack only appears in the time complexity of their algorithm as a multiplicative factor). The starting point of their result is the tight connection between approximation of the independence polynomial at positive activities and sampling from the associated *Gibbs distribution*; they then exploit this connection by showing that a natural Markov chain can sample efficiently from the Gibbs distribution. Their proof uses the connection between correlation decay and the mixing properties of Markov chains for the Gibbs distribution in a novel interesting fashion.

Unfortunately, to the best of our knowledge, no natural analogue of the Gibbs distribution is known for the negative activities setting. Thus, it remains an open problem to find if, and how, sampling techniques such as Markov chain Monte Carlo can be brought to bear upon the above question. Moreover, for applications to the LLL in the variable model, one would also need to remove the large girth assumption that appears to be crucial to the result of Efthymiou et al. [11].

A The Lovász Local Lemma and Shearer's Lemma

The Lovász Local Lemma (LLL) is a fundamental tool used in combinatorics to argue that the probability that none of a set of suitably constrained bad events occurs is positive. In abstract terms, the lemma is formulated in terms of n events $\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_n$ and a probability distribution μ on the events. However, only two pieces of data about the distribution μ are used in the formulation of lemma:

- The marginal probabilities $p_i := \mu(\mathcal{E}_i)$ of the events, and
- A *dependency graph* $G = (V, E)$ associated with μ . The vertices V are identified with the events

$\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_n$, and the graph is interpreted as stipulating that under the distribution μ , the event \mathcal{E}_i is independent of its non-neighbors in the graph G .

The original LLL [12, 41] provides *sufficient* conditions on the p_i and the dependency graph G that ensure that $\mu(\bigcap_{i=1}^n \mathcal{E}_i) > 0$, and hence $\bigcap_{i=1}^n \mathcal{E}_i \neq \emptyset$.

Define the set

$$\mathcal{L} = \{\mathbf{p} \in [0, 1]^n : \exists \mathbf{x} \in (0, 1)^n \text{ s.t. } p_i \leq x_i \prod_{j \in \Gamma(i)} (1 - x_j)\}.$$

Theorem A.1 (The Lovász Local Lemma [12, 41]). If $\mathbf{p} \in \mathcal{L}$ then $\mu(\bigcap_{i=1}^n \mathcal{E}_i) > 0$.

Shearer’s remarkable lemma [35] provides *necessary and sufficient* conditions for $\mu(\bigcap_{i=1}^n \mathcal{E}_i) > 0$ to hold for a given dependency graph G and vector of probabilities $\mathbf{p} \in [0, 1]^V$. Scott and Sokal [34] showed that Shearer’s conditions can be expressed very succinctly in the language of partition functions. Recall the definition of \mathcal{S} from Section 1.1.

Theorem A.2 (Shearer’s Lemma [34, 35]). If $\mathbf{p} \in \mathcal{S}$ then $\mu(\bigcap_{i=1}^n \mathcal{E}_i) > 0$.

Conversely, if $\mathbf{p} \notin \mathcal{S}$ then there exists a distribution μ with dependency graph G satisfying $\mu(\mathcal{E}_i) = p_i$ such that $\mu(\bigcap_{i=1}^n \mathcal{E}_i) = 0$.

The fact that $\mathcal{L} \subseteq \mathcal{S}$ follows indirectly from the statements of Theorems A.1 and A.2, but a direct argument is also known [21, Corollary 5.37].

A.1 The Variable Model

In order to design algorithmic forms of the LLL, some assumption must be made on the probability space. The most natural assumption is the “variable model”, in which the probability space consists of m independent variables with finite domain, each event depends on some subset of the variables, and that two events are adjacent in G if there is a common variable on which both depend.

This paper will focus on the specific scenario where the variables take values in $\{0, 1\}$. Concretely, the probability space is supported on $\Omega = \{0, 1\}^m$. Its distribution is $\mu_{\mathbf{z}}$, the product distribution on Ω with marginal vector \mathbf{z} . That is, if ω has distribution $\mu_{\mathbf{z}}$ then $\mathbb{E}[\omega] = \mathbf{z}$. Each event is a Boolean function of ω that only depends on certain coordinates $A_i \subseteq [m]$. Let G be the dependency graph on $V = [n]$ where i and j are adjacent if $A_i \cap A_j \neq \emptyset$. Since $\mu_{\mathbf{z}}$ is a product distribution, each event \mathcal{E}_i is clearly independent from its non-neighbors in G .

B Proofs from Section 2

Proof of Claim 2.1. From eq. (2.2), we have, for every $1 \leq i \leq n$, $1 - r_{S_i, v_i} = \frac{\check{q}_{S_i}}{\check{q}_{S_{i+1}}}$. Multiplying these equations,

we get

$$\prod_{i=1}^n (1 - r_{S_i, v_i}) = \frac{\check{q}_{S_1}}{\check{q}_{S_{n+1}}} = \frac{\check{q}_V}{\check{q}_\emptyset}.$$

Since $\check{q}_\emptyset = 1$, this yields the claim. \square

Proof of Lemma 2.3. Define $S_i = S \setminus \{u, v_1, v_2, \dots, v_{i-1}\}$ for $1 \leq i \leq k$. From eq. (2.2), we then have, for $1 \leq i \leq k$, $\frac{1}{1 - r_{S_i, v_i}} = \frac{\check{q}_{S_{i+1}}}{\check{q}_{S_i}}$. Multiplying these equations, we get

$$\prod_{i=1}^k \frac{1}{1 - r_{S_i, v_i}} = \frac{\check{q}_{S_{k+1}}}{\check{q}_{S_1}} = \frac{\check{q}_{S \setminus \Gamma^+(u)}}{\check{q}_{S \setminus \{u\}}},$$

where in the last equation we use $S_{k+1} = S \setminus \Gamma^+(u)$ and $S_1 = S \setminus \{u\}$. (Recall that $\Gamma^+(u)$ is the set containing u and all its neighbors in G). The claim of the lemma now follows since $r_{S, u} = p_u \cdot \frac{\check{q}_{S \setminus \Gamma^+(u)}}{\check{q}_{S \setminus \{u\}}}$ (see eq. (2.2)). \square

Proof of Lemma 2.7. This is a special case of Corollary 2.27 (b) of Scott and Sokal [34]. See also [21, Section 5.3]. \square

Proof of Lemma 2.8. From eq. (2.2) we have $r_{S, u} = \frac{p_u \check{q}_{S \setminus \Gamma^+(u)}}{\check{q}_{S \setminus \{u\}}} = 1 - \frac{\check{q}_S}{\check{q}_{S \setminus \{u\}}}$. From Lemma 2.7, we have $0 < \check{q}_V \leq \check{q}_S \leq \check{q}_{S \setminus \{u\}} \leq \check{q}_{S \setminus \Gamma^+(u)}$, which yields the claim. \square

C Inequalities in the complex plane

Finally, we enumerate some simple inequalities that are used in our proofs.

Fact C.1. Let z be a complex number such that $|z| \leq \tau < 1$. Then $\frac{1}{|1-z|} \leq \frac{1}{1-\tau}$.

Proof. $|1-z| \geq 1 - |z| \geq 1 - \tau$, which implies the claim since $\tau < 1$. \square

Fact C.2. Let $(x_i)_{i=1}^n$ and $(y_i)_{i=1}^n$ be two sequences of complex numbers with the y_i non-zero such that

$$(C.1) \quad \left| \frac{x_i}{y_i} - 1 \right| \leq \varepsilon,$$

where $\varepsilon \leq 1/n$. Then, we have

$$\left| \prod_{i=1}^n x_i - \prod_{i=1}^n y_i \right| \leq 2n\varepsilon \cdot \prod_{i=1}^n |y_i|.$$

Proof. For each i , define z_i so that $x_i = y_i(1 + z_i)$. Note

that $|z_i| \leq \varepsilon$ for each i . We therefore have

$$\begin{aligned} \left| \prod_{i=1}^n x_i - \prod_{i=1}^n y_i \right| &= \prod_{i=1}^n |y_i| \left| \prod_{i=1}^n (1 + z_i) - 1 \right| \\ &\leq \sum_{i=1}^n \binom{n}{i} \varepsilon^i \cdot \prod_{i=1}^n |y_i| \\ &\leq n\varepsilon \sum_{i=1}^n \frac{(n\varepsilon)^{i-1}}{i!} \cdot \prod_{i=1}^n |y_i|, \text{ using } \binom{n}{i} \leq \frac{n^i}{i!}, \\ &\leq \prod_{i=1}^n |y_i| \cdot n\varepsilon \sum_{i=1}^n \frac{1}{i!} \\ &\leq n\varepsilon \cdot (e-1) \cdot \prod_{i=1}^n |y_i| \leq 2n\varepsilon \cdot \prod_{i=1}^n |y_i|, \end{aligned}$$

where the last line uses the fact that $n\varepsilon < 1$. \square

We will also need the following consequence of the mean value theorem. Fix a complex number λ and a positive integer $d > 0$ and let $f(\mathbf{x}) = f(x_1, x_2, \dots, x_d)$ be defined as

$$f(x_1, x_2, \dots, x_d) := \lambda \prod_{i=1}^d \frac{1}{1 - x_i}$$

defined when $|x_i| < 1$ for all i .

Theorem C.3 (Mean value theorem). Let $\mathbf{x} = (x_1, x_2, \dots, x_d)$ and $\mathbf{y} = (y_1, y_2, \dots, y_d)$ be two sequences of complex numbers and let $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \dots, \gamma_d)$ be such that $|x_i|, |y_i| \leq \gamma_i < 1$ for $1 \leq i \leq d$. Then

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq |f(\boldsymbol{\gamma})| \sum_{i=1}^d \frac{|x_i - y_i|}{1 - \gamma_i}.$$

Proof. Let $g(t) := f(t\mathbf{x} + (1-t)\mathbf{y})$ for $t \in [0, 1]$. Note that g is continuously differentiable on its domain (since $|x_i|, |y_i| < 1$). Hence $|g'(t)|$ attains its maximum at some point $t_0 \in [0, 1]$. Let $\mathbf{z} = t_0\mathbf{x} + (1-t_0)\mathbf{y}$. Note that $|z_i| \leq \gamma_i$ for all i . We now have

$$\begin{aligned} |f(\mathbf{x}) - f(\mathbf{y})| &= |g(1) - g(0)| \leq \int_0^1 |g'(t)| dt \leq |g'(t_0)| \\ &= |f(\mathbf{z})| \left| \sum_{i=1}^d \frac{y_i - x_i}{1 - z_i} \right| \\ &\leq |f(\boldsymbol{\gamma})| \sum_{i=1}^d \frac{|x_i - y_i|}{1 - \gamma_i}, \end{aligned}$$

where the last line uses the form of f and Fact C.1. \square

D Proofs from Section 4

D.1 Correctness of the rounding procedure

Define the following set

$$\mathcal{R} = \{ \mathbf{p} \in [0, 1]^n : \check{q}_V(\mathbf{p}) > 0 \}.$$

Lemma D.1. Let $f : [0, 1] \rightarrow \mathcal{R}$ be a continuous function with $f(0) \in \mathcal{S}$. Then $f(t) \in \mathcal{S}$ for all $t \in [0, 1]$.

Proof. Suppose that $\{t \in [0, 1] : f(t) \notin \mathcal{S}\}$ is non-empty, and let τ be the infimum of that set. By continuity of f and the fact that \mathcal{S} is open, we must have $f(\tau) \notin \mathcal{S}$. That is, there exists $S \subseteq V$ with $0 \geq \check{q}_S(f(\tau))$. As $f(0) \in \mathcal{S}$, we have $\tau > 0$. The range of f is \mathcal{R} , so $\check{q}_V(f(\tau)) > 0 \geq \check{q}_S(f(\tau))$. For sufficiently small $\varepsilon > 0$, continuity implies that $\check{q}_V(f(\tau - \varepsilon)) > \check{q}_S(f(\tau - \varepsilon))$. But, by definition of τ , we have $f(\tau - \varepsilon) \in \mathcal{S}$. This contradicts Lemma 2.7. \square

Consider rounding the coordinate z_i . Let $\mathbf{z}' \rightarrow \mathbf{z}$. If $\frac{\partial \check{q}_V}{\partial z_i}(\mathbf{p}(\mathbf{z})) \geq 0$ then set $z'_i \leftarrow 1$, otherwise set $z'_i \leftarrow 0$. Then, by multilinearity, for all \mathbf{y} on the line segment between \mathbf{z} and \mathbf{z}' we have $\check{q}_V(\mathbf{y}) \geq \check{q}_V(\mathbf{z}) > 0$, and hence $\mathbf{y} \in \mathcal{R}$. Lemma D.1 now implies that $\mathbf{p}(\mathbf{z}') \in \mathcal{S}$.

D.2 Impossibility of rounding within \mathcal{L}

Consider the scenario $\Omega = \{0, 1\}^{15}$ with $\mathbf{z} = (1/2, \dots, 1/2)$ so that $\mu_{\mathbf{z}}$ is the uniform distribution. Define the events

$$\begin{aligned} \mathcal{E}_1 &= \{ \boldsymbol{\omega} : \sum_{i=1}^6 \omega_i \in \{0, 2, 6\} \} \\ \mathcal{E}_2 &= \{ \boldsymbol{\omega} : \omega_6 = \omega_7 = \omega_8 = 0 \} \\ \mathcal{E}_3 &= \{ \boldsymbol{\omega} : \omega_8 = \omega_9 = \omega_{10} = 1 \} \\ \mathcal{E}_4 &= \{ \boldsymbol{\omega} : \sum_{i=10}^{15} \omega_i \in \{0, 2, 6\} \}. \end{aligned}$$

The dependency graph G for these events is the path graph 1-2-3-4.

Recalling that $p_i = \mu_{\mathbf{z}}(\mathcal{E}_i)$, we have

$$\mathbf{p} = \left(\frac{17}{64}, \frac{1}{8}, \frac{1}{8}, \frac{17}{64} \right).$$

One may verify that $\mathbf{p} \in \mathcal{L}$ using the vector

$$\mathbf{x} = \left(\frac{4}{10}, \frac{3}{10}, \frac{3}{10}, \frac{4}{10} \right).$$

Suppose we round z_3 to 0. (The other case is symmetric.) The probability vector is now

$$\mathbf{p}' = \left(\frac{17}{64}, \frac{1}{4}, 0, \frac{17}{64} \right).$$

We claim that $\mathbf{p}' \notin \mathcal{L}$. To see this, note that $p_3 = 0$, so we may delete vertex 3, after which the dependency graph has the single edge 1-2. For a graph consisting of a single edge, the region \mathcal{L} can be shown to be precisely $\{ (p_1, p_2) : \sqrt{p_1} + \sqrt{p_2} \leq 1 \}$. Since this condition is violated for \mathbf{p}' , we have $\mathbf{p}' \notin \mathcal{L}$.

Thus z_3 cannot be rounded either to 0 or to 1 while preserving membership in \mathcal{L} .

E Hardness of evaluation and deciding membership

In this section we complement our positive results with some negative ones. The first such hardness results show that exact evaluation of \check{q}_V and exact decision of membership in Shearer region are both #P-hard: due to lack of space these results are consigned to the full version and we focus here on results about approximation. In Section E.1 we show that approximation within exponentially small error is computationally hard, and then deduce that algorithms for approximating \check{q}_V must have runtime that depends on the slack. Finally, we can also show a positive result for a relaxation of the membership problem – that one can efficiently decide membership in the region for the original LLL, which is a strict subset of Shearer’s region – but due to lack of space, this result is also not included here and may be found in the full version of the paper.

E.1 Approximate evaluation and membership

As our main positive result shows, $\check{q}_G(p)$ for p well inside the Shearer region (with constant slack) can indeed be evaluated *approximately*, within polynomially small error. The proofs of some of these results are not included here due to lack of space and may be found in the full version.

Theorem E.1. For a 4-regular graph $G = (V, E)$ and $|V| < k < |V|^2$ given on the input, it is #P-hard to compute $\check{q}_G(1/k, \dots, 1/k)$ within an additive error of $1/(2k^{|V|/3})$.

Theorem E.2. For a graph $G = (V, E)$ and rational $p \in [0, 1]^V$ given on the input, it is #P-hard to distinguish whether $\check{q}_G(p) \geq 1/|V|^{|V|}$ or $\check{q}_G(p) \leq 0$.

Theorem E.3. For a graph $G = (V, E)$ and rational $(p_1, \dots, p_n) \in [0, 1]^V$ given on the input, it is #P-hard to distinguish between $(p_1 + \varepsilon, \dots, p_n + \varepsilon) \in \mathcal{S}_G$ and $(p_1, \dots, p_n) \notin \mathcal{S}_G$, for $\varepsilon = 1/|V|^{|V|}$.

It remains open whether membership in the Shearer region is polynomially checkable within *polynomially small error*.

Next, we use Theorem E.3 to prove that it is in fact #P-hard to approximate the independence polynomial even within *polynomially large factors*, when \mathbf{p} is inside but close to the boundary of the Shearer region.

Theorem E.4. For a graph $G = (V, E)$, $|V| = n$, and rational $\mathbf{z} \in [0, 1]^V$ given on the input, such that $(1 + \frac{1}{n^{2n}})\mathbf{z} \in \mathcal{S}_G$, it is #P-hard to approximate $\check{q}_V(\mathbf{z})$ with any poly(n) factor.

Proof. Suppose that given G, \mathbf{z} as above, we can compute a number \check{Q}_V such that $\check{q}_V(\mathbf{z}) \leq \check{Q}_V \leq n^c \check{q}_V(\mathbf{z})$, for some

absolute constant $c > 0$. Then clearly we can also do this for $\check{q}_S(\mathbf{z})$, $S \subseteq V$, by considering the subgraph induced by S . Suppose also that n is sufficiently large, say $n \geq 2c + 2$. We claim that then by a polynomial number of calls to such an algorithm, we can distinguish for a given point $\mathbf{p} \in [0, 1]^V$ whether $\mathbf{p} + \frac{1}{n^c} \mathbf{1} \in \mathcal{S}_G$ or $\mathbf{p} \notin \mathcal{S}_G$, which is a #P-hard problem by Theorem E.3.

Let $\phi(t) = \check{q}_V(t\mathbf{p})$. Clearly $\phi(0) = 1$, and it was shown in [21] that ϕ is convex and decreasing. We aim to find the minimum $t > 0$ such that $\phi(t) = 0$, which defines the nearest point on the boundary of \mathcal{S}_G in the direction of \mathbf{p} . We can assume that $\sum_{i=1}^n p_i \geq 1$, otherwise $\mathbf{p} \in \mathcal{S}_G$ trivially. We use the following algorithm: We start with $t = 0$. Given t , we estimate $\phi(t)$ and $\phi'(t)$ (within polynomial factors) using the assumed algorithm. This can be done, since $\phi(t) = \check{q}_V(t\mathbf{p})$, and

$$\phi'(t) = \frac{d}{dt} \check{q}_V(t\mathbf{p}) = \sum_{i=1}^n p_i \frac{\partial}{\partial z_i} \check{q}_V(\mathbf{z}) \Big|_{\mathbf{z} = t\mathbf{p}} = - \sum_{i=1}^n p_i \check{q}_{V \setminus \Gamma^+(i)}(t\mathbf{p}).$$

We will show that we only apply this computation to points t such that $(1 + \frac{1}{n^{2n}})t\mathbf{p} \in \mathcal{S}$. For such points $\phi(t) > 0$, $\phi'(t) < 0$ and we can also estimate $\frac{\phi(t)}{|\phi'(t)|}$. Let $D(t)$ be our estimate, such that $n^{-2c} \frac{\phi(t)}{|\phi'(t)|} \leq D(t) \leq \frac{\phi(t)}{|\phi'(t)|}$. Given this estimate, we replace t by $t' = t + \frac{1}{2}D(t)$. We repeat this process as long as $D(t) \geq 1/n^{n+1+2c}$ and $t < 1$. If we reach $t \geq 1$, we answer YES; else if $D(t)$ drops below $1/n^{n+1+2c}$, we answer NO.

We note the following: Assuming that the minimum positive root of ϕ is ξ_0 and $0 \leq t < \xi_0$, we have $t + D(t) \leq t + \frac{\phi(t)}{|\phi'(t)|} \leq \xi_0$ by convexity of ϕ . Therefore, the additive slack at any point t is at least $D(t)$. Since we update the point to $t' = t + \frac{1}{2}D(t)$, we always retain slack at least $\frac{1}{2}D(t)$, which is guaranteed to be at least $\frac{1}{2n^{n+1+2c}} \geq \frac{1}{n^{2n}}$ (for $n \geq 2c + 2$), otherwise we terminate. This proves the above claim that we only evaluate at points t such that $(1 + \frac{1}{n^{2n}})t\mathbf{p} \in \mathcal{S}$.

On the other hand, if $\delta := \xi_0 - t$, we have $\check{q}_V(t\mathbf{p} + \delta p_i \mathbf{e}_i) \geq 0$ for $1 \leq i \leq n$ since $(t + \delta)\mathbf{p}$ is at the boundary of \mathcal{S}_G . We then have

$$\begin{aligned} \phi(t + \delta) - \phi(t) &\leq \min_{1 \leq i \leq n} \check{q}_V(t\mathbf{p} + \delta p_i \mathbf{e}_i) - \check{q}_V(t\mathbf{p}) \\ &= - \max_{1 \leq i \leq n} \delta p_i \frac{\partial \check{q}_V}{\partial z_i} \Big|_{\mathbf{z} = t\mathbf{p}} \\ &\leq - \frac{\delta}{n} \sum_{i=1}^n p_i \frac{\partial \check{q}_V}{\partial z_i} \Big|_{\mathbf{z} = t\mathbf{p}} = \frac{\delta}{n} \phi'(t). \end{aligned}$$

Therefore, since $\phi(t + \delta) = 0$, we get $\frac{\phi(t)}{|\phi'(t)|} \geq \frac{\delta}{n} = \frac{\xi_0 - t}{n}$. By our approximation guarantee, $D(t) \geq n^{-2c} \frac{\phi(t)}{|\phi'(t)|} \geq n^{-2c-1}(\xi_0 - t)$. (Note that this also means that $(t + n^{1+2c}D(t))\mathbf{p} \notin \mathcal{S}$.) So when we replace t by $t + \frac{1}{2}D(t)$, we decrease the distance $\xi_0 - t$ to the nearest

root by a factor of $1 - 1/(2n^{2c+1})$ in the worst case. After $2n^{2c+1}(n+2+2c)\log n$ steps, the distance decreases by a factor of

$$\left(1 - \frac{1}{2n^{2c+1}}\right)^{2n^{2c+1}(n+2+2c)\log n} < \frac{1}{n^{n+2+2c}}.$$

Initially, we have $\xi_0 \leq n$ because $p_i \geq 1/n$ for some $i \in [n]$. Hence, the quantity $\xi_0 - t$ as well as $D(t)$ must shrink below $1/n^{n+1+2c}$ in a polynomial number of steps.

If we terminate because $t \geq 1$, we have certified that $\mathbf{p} \in \mathcal{S}$ and we can answer YES. If we terminate because $D(t) < 1/n^{n+1+2c}$ then it is the case that $t < 1$, and we know that $(t+n^{1+2c}D(t))\mathbf{p} \notin \mathcal{S}$. Hence $(1+1/n^n)\mathbf{p} \notin \mathcal{S}$ and we can answer NO. \square

Corollary E.5 (restatement of Theorem 1.5). If there is an algorithm to estimate $\check{q}_V(\mathbf{p})$ to within a poly(n) multiplicative factor, assuming that $(1+\alpha)\mathbf{p} \in \mathcal{S}$, and running in time $(n \log \frac{1}{\alpha})^{O(\log n)}$, then $\#\mathbf{P} \subseteq \text{DTIME}(n^{O(\log n)})$.

Proof. Suppose we have such an algorithm. Then we can run it for $\alpha = \frac{1}{n^{2n}}$, and solve a $\#\mathbf{P}$ -hard problem (from Theorem E.4) in running time $(n \log \frac{1}{\alpha})^{O(\log n)} = n^{O(\log n)}$. \square

In contrast, the running time of our algorithm (for a constant-factor approximation) is $n^{O(\frac{1}{\alpha} \log d)}$. Again, there is an open question here, whether there is an approximation algorithm (possibly even an FPTAS) under these conditions with running time at most quasi-poly($n, 1/\alpha$).

F Optimality of the decay rate

We saw above that when the probability vector \mathbf{p} input to our approximation algorithm (Algorithm 1) for the independence polynomial has slack α (i.e., when $(1+\alpha)\mathbf{p} \in \mathcal{S}$), the running time of the algorithm is exponential in $1/\sqrt{\alpha}$. In the applications outlined in Sections 4 and 5, Algorithm 1 is invoked with \mathbf{p} that have slack $\Theta(1/n)$ (where n is the number of vertices in the graph), and the above exponential dependence on $1/\sqrt{\alpha}$ leads to a sub-exponential dependence on n . While this is qualitatively superior to exponential time algorithms that would result from naive brute force methods, or even methods based on approximate counting algorithms whose running times are exponential in $1/\alpha$ (e.g., those in [31]), one might still ask if it is possible to get a better dependence on α so as to improve the running times obtained in our applications.

In this section, we present evidence that ‘‘correlation-decay’’ based methods based on Weitz’s framework cannot in fact break the sub-exponential barrier. To do this, we revisit Theorem 3.7, which is the main ingredient in the complexity analysis of Algorithm 1. Recall that Theorem 3.7 considers the effect of truncating the exponentially

large computation tree (generated by Lemma 2.3) at a finite depth ℓ by showing that the tree-recurrence of eq. (2.4) causes an amortized version of the error to decay by a factor $(1 - \Omega(\sqrt{\alpha}))$ at each level of the computation tree. The factor of $\sqrt{\alpha}$ in the running time of Algorithm 1 came precisely from this form of the decay factor. In particular, to show that one cannot do better than a sub-exponential dependence on $1/\alpha$ in the running time in this framework, we need to show that a decay rate better than $(1 - \Omega(\sqrt{\alpha}))$, where c is an appropriate constant in $(0, 1/2]$, cannot be obtained in the general case.

To establish this, we consider the case where the computation tree is actually a d -ary tree. In addition to being the simplest possible example of a computation tree, the d -ary tree also arises as the truncated computation tree when Algorithm 1 is applied to a vertex in a locally tree like graph (e.g., a random vertex in a random regular graph).

For a given activity λ , the tree recurrence on a d -ary tree reduces to the following univariate recurrence:

$$f(x) := \frac{\lambda}{(1-x)^d}.$$

Since the d -ary tree has degree $d+1$ at all vertices (except at the root which has degree d), the radius of the univariate Shearer region for it is $\lambda'_c(d+1) = \frac{d^d}{(d+1)^{d+1}}$. Note that f and all its derivatives are increasing functions of $x \in (0, 1)$. It is also not hard to show that when $0 < \lambda < \lambda'_c(d+1)$, this recurrence has two fixed points $x^* < x^\dagger$ in $(0, 1)$, such that $\lim_{\ell \rightarrow \infty} f^\ell(x) = x^*$. Our goal now is to show that when λ has slack α (i.e., $\lambda = (1-\alpha)\lambda'_c(d+1)$), the rate of convergence of this recurrence is no better than $(1 - O(\sqrt{\alpha}))$. In particular, we will show that

$$\left|f'(0) - x^*\right| \geq \Omega\left((1 - O(\sqrt{\alpha}))^\ell\right).$$

More formally, we have the following theorem.

Theorem F.1. Let d be a large enough positive degree. Let $\alpha \in (0, 1/8)$ be arbitrary, and set $\lambda = (1-\alpha)\lambda'_c(d+1)$. Then, there exist $c_0 = c_0(\alpha, d), c_1 = c_1(\alpha, d)$ and $l_0 = l_0(\alpha, d)$ such that for all $l \geq l_0$,

$$\left|f^l(0) - x^*\right| \geq c_0 (1 - c_1 \sqrt{\alpha})^{l-l_0}.$$

Before proving the theorem, we record some useful properties of the recurrence f .

Observation F.2. f and all its derivatives are strictly increasing in λ and x when $x \in [0, 1)$.

Proof. This follows from the form of f . \square

Fact F.3. Given $\lambda > 0$ and $d \geq 3$, let $x^*(d, \lambda)$ be the smallest fixed point (if one exists) of f in $(0, 1)$. Then, for all $d \geq 3$, $x^*(d, \lambda'_c(d+1)) = \frac{1}{d+1}$ and $f'(x^*(d, \lambda'_c(d+1))) = 1$. Further, $x^*(d, \lambda)$ and $f'(x^*(d, \lambda))$ are strictly increasing functions of λ for $\lambda < \lambda'_c(d+1)$.

Proof. Since f is convex in $[0, 1)$ and satisfies both $f(0) > 0$ and $\lim_{t \uparrow 1} f(t) - t = \infty$, it follows that the equation $x = f(x)$ has at most two roots in $[0, 1)$. Further, since $\lambda < \lambda'_c(d+1) \leq 1$, we also have that $f'(x) \leq 1$ at the smaller such root. It then follows that if $f'(x) = 1$ at such a root x , then x is the unique root of f in $[0, 1)$. We then verify by direct calculation that $f_{d, \lambda'_c(d+1)}(1/(d+1)) = 1/(d+1)$ and $f'_{d, \lambda'_c(d+1)}(1/(d+1)) = 1$.

Since f is strictly increasing in λ , it then follows that $x^*(d, \lambda)$ is also a strictly increasing function of λ . Since f' is strictly increasing in both λ and x the last claim also follows. \square

We are now ready to prove Theorem F.1.

Proof of Theorem F.1. It follows from Fact F.3 that $f'(x^*(d, \lambda)) < 1$. So let $\delta \in (0, 1)$ be such that $1 - \delta = f'(x^*(d, \lambda))$. Since we also have $f(x^*(d, \lambda)) = x^*(d, \lambda)$, we can solve to get $x^*(d, \lambda) = \frac{1-\delta}{d+1-\delta}$, and hence $\lambda = (1-\delta) \cdot \frac{d^d}{(d+1-\delta)^{d+1}}$. Since $\lambda = (1-\alpha)\lambda'_c(d+1)$, we have

$$(F.1) \quad 1 - \alpha = (1 - \delta) \left(1 - \frac{\delta}{d+1}\right)^{-(d+1)}.$$

We now use the following inequalities which are valid for all large enough positive d :

$$(F.2) \quad \forall x \in [0, 1], (1-x) \left(1 - \frac{x}{d+1}\right)^{-(d+1)} \leq 1 - x^3$$

$$(F.3) \quad \exists \mu > 0 \text{ such that } \forall x \in [0, 1/2],$$

$$(1-x) \left(1 - \frac{x}{d+1}\right)^{-(d+1)} \leq 1 - (x/\mu)^2.$$

Inequality (F.2) applied to eq. (F.1) implies that $\delta \leq \alpha^{1/3}$ for all $\alpha \in (0, 1)$. Thus, for $\alpha \in (0, 1/8)$, $\delta \in (0, 1/2)$. We can thus apply inequality (F.3) to eq. (F.1) for such α to get $\delta \leq \mu\sqrt{\alpha}$ when $\alpha \in (0, 1/8)$, where $\mu = \mu(d)$ is the constant appearing in eq. (F.3). Thus, $1 - \mu\sqrt{\alpha} \leq f'(x^*(d, \lambda)) \leq 1$. Further, by the continuity and monotonicity of f' , there exists an $\varepsilon > 0$ such that

$$(F.4) \quad x \in [x^*(d, \lambda) - \varepsilon, x^*(d, \lambda)] \implies f'(x) \geq 1 - 2\mu\sqrt{\alpha}.$$

Now, $(f^l(0))_{l=0}^\infty$ is a strictly increasing sequence lying strictly below the fixed point $x^*(d, \lambda)$, and further satisfies

$$\begin{aligned} x^*(d, \lambda) - f^{l+1}(0) &= f(x^*(d, \lambda)) - f(f^l(0)) \\ &\leq (1 - \delta)(x^*(d, \lambda) - f^l(0)). \end{aligned}$$

It therefore follows that there is an l_0 such that for $l \geq l_0$, $f^l(0) \in [x^*(d, \lambda) - \varepsilon, x^*(d, \lambda)]$. For $l \leq l_0$ we therefore have

$$\begin{aligned} x^*(d, \lambda) - f^{l+1}(0) &= f(x^*(d, \lambda)) - f(f^l(0)) \\ &\geq (1 - 2\mu\sqrt{\alpha})(x^*(d, \lambda) - f^l(0)), \end{aligned}$$

where the last inequality uses eq. (F.4). The result now follows by an induction on l . \square

References

- [1] ACHLIOPTAS, D., AND ILIOPOULOS, F. Random walks that find perfect objects and the Lovász local lemma. In *Proc. 55th IEEE Symp. on Foundations of Computer Science (FOCS)* (2014), IEEE, pp. 494–503.
- [2] ALON, N. A parallel algorithmic version of the local lemma. *Random Structures and Algorithms* 2, 4 (1991), 367–378.
- [3] BARVINOK, A. Computing the partition function for cliques in a graph. *Theory of Computing* 11, 13 (2015), 339–355.
- [4] BARVINOK, A. Computing the permanent of (some) complex matrices. *Found. Comput. Math.* 16, 2 (Jan. 2015), 329–342. [arXiv:1405.1303](https://arxiv.org/abs/1405.1303).
- [5] BARVINOK, A., AND SOBERÓN, P. Computing the partition function for graph homomorphisms. *Combinatorica* (May 2016). Available online at <http://dx.doi.org/10.1007/s00493-016-3357-2>.
- [6] BARVINOK, A., AND SOBERÓN, P. Computing the partition function for graph homomorphisms with multiplicities. *J. Combin. Theory, Ser. A* 137 (Jan. 2016), 1–26.
- [7] BAYATI, M., GAMARNIK, D., KATZ, D., NAIR, C., AND TETALI, P. Simple Deterministic Approximation Algorithms for Counting Matchings. In *Proc. 39th ACM Symp. on Theory of Computing (STOC)* (2007), ACM, pp. 122–127.
- [8] BECK, J. An algorithmic approach to the Lovász local lemma. I. *Random Structures and Algorithms* 2, 4 (1991), 343–365.
- [9] CAI, J.-Y., AND CHEN, X. Complexity of counting CSP with complex weights. In *Proc. 44th ACM Symp. on Theory of Computing (STOC)* (2012), ACM, pp. 909–920.
- [10] CHANDRASEKARAN, K., GOYAL, N., AND HAEUPLER, B. Deterministic algorithms for the Lovász local lemma. *SIAM J. Comput.* 42, 6 (2013).
- [11] EFTHYMIU, C., HAYES, T. P., ŠTEFANKOVIČ, D., VIGODA, E., AND YIN, Y. Convergence of MCMC and loopy BP in the tree uniqueness region for the hard-core model. In *Proc. 57th IEEE Symp. on Foundations of Computer Science (FOCS)* (2016), IEEE Computer Society, pp. 704–713.
- [12] ERDŐS, P., AND LOVÁSZ, L. Problems and results on 3-chromatic hypergraphs and some related questions. In *Infinite and finite sets*, A. H. et al., Ed., vol. 10 of *Colloquia Mathematica Societatis János Bolyai*. North-Holland, Amsterdam, 1975, pp. 609–628.
- [13] GALANIS, A., GE, Q., ŠTEFANKOVIČ, D., VIGODA, E., AND YANG, L. Improved inapproximability results for counting independent sets in the hard-core model. *Random Struct. Algorithms* 45, 1 (2014), 78–110.
- [14] GALANIS, A., GOLDBERG, L. A., AND ŠTEFANKOVIČ, D. Inapproximability of the Independent Set Polynomial Below the Shearer Threshold. In *Proc. 44th International Colloquium on Automata, Languages, and Programming*

- (ICALP) (2017), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 28:1–28:13. [arXiv:1612.05832](https://arxiv.org/abs/1612.05832).
- [15] GALANIS, A., ŠTEFANKOVIČ, D., AND VIGODA, E. Inapproximability for Antiferromagnetic Spin Systems in the Tree Nonuniqueness Region. *J. ACM* 62, 6 (Dec. 2015), 50:1–50:60. [arXiv:1305.2902](https://arxiv.org/abs/1305.2902).
- [16] GAMARNIK, D., AND KATZ, D. Correlation decay and deterministic FPTAS for counting list-colorings of a graph. In *Proc. 18th ACM-SIAM Symp. on Discrete Algorithms (SODA)* (2007), SIAM, pp. 1245–1254.
- [17] HAMMERSLEY, J. M. Percolation processes II. The connective constant. *Math. Proc. Cambridge Philosophical Soc.* 53, 03 (1957), 642–645.
- [18] HARRIS, D. G. *Algorithms and Generalizations for the Lovász Local Lemma*. PhD thesis, University of Maryland, 2015.
- [19] HARVEY, N. J. A., SRIVASTAVA, P., AND VONDRÁK, J. Computing the independence polynomial in Shearer’s region for the LLL, Aug. 2016. [arXiv:1608.02282v1](https://arxiv.org/abs/1608.02282v1).
- [20] HARVEY, N. J. A., AND VONDRÁK, J. An algorithmic proof of the Lovász local lemma via resampling oracles. In *Proc. 56th IEEE Symp. on Foundations of Computer Science (FOCS)* (2015), IEEE, pp. 1327–1346.
- [21] HARVEY, N. J. A., AND VONDRÁK, J. An algorithmic proof of the Lovász local lemma via resampling oracles, 2015. [arXiv:1504.02044](https://arxiv.org/abs/1504.02044).
- [22] HOLROYD, A. E., AND LIGGETT, T. M. Finitely dependent coloring. *Forum of Mathematics, Pi* 4 (2016). [arXiv:1403.2448](https://arxiv.org/abs/1403.2448).
- [23] KOLIPAKA, K., AND SZEGEDY, M. Moser and Tardos meet Lovász. In *Proc. 43rd ACM Symp. on Theory of Computing (STOC)* (2011), ACM, pp. 235–244.
- [24] LEE, T. D., AND YANG, C. N. Statistical theory of equations of state and phase transitions. II. Lattice gas and Ising model. *Phys. Rev.* 87, 3 (Aug. 1952), 410–419.
- [25] LI, L., LU, P., AND YIN, Y. Correlation decay up to uniqueness in spin systems. In *Proc. 24th ACM-SIAM Symp. on Discrete Algorithms (SODA)* (2013), SIAM, pp. 67–84.
- [26] LIU, J., AND LU, P. FPTAS for #BIS with Degree Bounds on One Side. In *Proc. 47th ACM Symp. on Theory of Computing (STOC)* (2015), ACM, pp. 549–556.
- [27] LIU, J., AND LU, P. FPTAS for counting monotone CNF. In *Proc. 26th ACM-SIAM Symp. on Discrete Algorithms (SODA)* (2015), SIAM, pp. 1531–1548. [arXiv:1311.3728](https://arxiv.org/abs/1311.3728).
- [28] MADRAS, N., AND SLADE, G. *The Self-Avoiding Walk*. Birkhäuser, 1996.
- [29] MOSER, R., AND TARDOS, G. A constructive proof of the general Lovász Local Lemma. *J. ACM* 57, 2 (2010).
- [30] MOSER, R. A. A constructive proof of the Lovász local lemma. In *Proceedings of the ACM Symposium on Theory of Computation (STOC)* (2009).
- [31] PATEL, V., AND REGTS, G. Deterministic polynomial-time approximation algorithms for partition functions and graph polynomials, July 2016. [arXiv:1607.01167v2](https://arxiv.org/abs/1607.01167v2).
- [32] REGTS, G. Zero-free regions of partition functions with applications to algorithms and graph limits. *Combinatorica* (2015). To appear.
- [33] RESTREPO, R., SHIN, J., TETALI, P., VIGODA, E., AND YANG, L. Improved mixing condition on the grid for counting and sampling independent sets. *Probab. Theory Relat. Fields* 156, 1–2 (June 2013), 75–99. Extended abstract in Proc. IEEE Symp. on Foundations of Computer Science (FOCS), 2011.
- [34] SCOTT, A., AND SOKAL, A. The Repulsive Lattice Gas, the Independent-Set Polynomial, and the Lovász Local Lemma. *J. Stat. Phys.* 118, 5–6 (2004), 1151–1261.
- [35] SHEARER, J. B. On a problem of Spencer. *Combinatorica* 5, 3 (1985).
- [36] SINCLAIR, A., SRIVASTAVA, P., AND THURLEY, M. Approximation algorithms for two-state anti-ferromagnetic spin systems on bounded degree graphs. *J. Stat. Phys.* 155, 4 (2014), 666–686. A preliminary version appeared in *Proc. 23rd ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2012.
- [37] SINCLAIR, A., SRIVASTAVA, P., ŠTEFANKOVIČ, D., AND YIN, Y. Spatial mixing and the connective constant: Optimal bounds. *Probab. Theory Relat. Fields* 168, 1–2 (July 2017), 153–197. Available online at <http://dx.doi.org/10.1007/s00440-016-0708-2>. An extended abstract of this paper appeared in *Proc. 26th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2015, pp. 1549–1563. [arXiv:1410.2595](https://arxiv.org/abs/1410.2595).
- [38] SINCLAIR, A., SRIVASTAVA, P., AND YIN, Y. Spatial mixing and approximation algorithms for graphs with bounded connective constant. In *Proc. 54th IEEE Symp. on Foundations of Computer Science (FOCS)* (2013), IEEE Computer Society, pp. 300–309. Full version available at [arXiv:1308.1762v1](https://arxiv.org/abs/1308.1762v1). Improved versions of results in this paper appear in [37].
- [39] SLY, A. Computational transition at the uniqueness threshold. In *Proc. 51st IEEE Symp. on Foundations of Computer Science (FOCS)* (2010), IEEE, pp. 287–296.
- [40] SLY, A., AND SUN, N. Counting in two-spin models on d -regular graphs. *Ann. Probab.* 42, 6 (Nov. 2014), 2383–2416.
- [41] SPENCER, J. Asymptotic lower bounds for Ramsey functions. *Discrete Math.* 20 (1977), 69–76.
- [42] SRINIVASAN, A. Improved algorithmic versions of the Lovász local lemma. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2008).
- [43] SRIVASTAVA, P. Approximating the hard core partition function with negative activities, Apr. 2015. Available at <http://www.tifr.res.in/~piyush.srivastava/research.html#Notes>.
- [44] TODO, S. Transfer-matrix study of negative-fugacity singularity of hard-core lattice gas. *Int. J. Mod. Phys C* 10 (1999), 517–529. [arXiv:cond-mat/9703176](https://arxiv.org/abs/cond-mat/9703176).
- [45] VALIANT, L. G. The complexity of enumeration and reliability problems. *SIAM J. Comput.* 8, 3 (1979), 410–421.
- [46] WEITZ, D. Counting independent sets up to the tree threshold. In *Proc. 38th ACM Symp. on Theory of Computing (STOC)* (2006), ACM, pp. 140–149.
- [47] YANG, C. N., AND LEE, T. D. Statistical theory of equations of state and phase transitions. I. Theory of condensation. *Phys. Rev.* 87, 3 (Aug. 1952), 404–409.