Combinatorics, Probability and Computing (2026), 1–33
doi:10.1017/S0963548326100558

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

Computing volume of a polytope and integrals in high dimensions deterministically

David Gamarnik[1,2,3]* and Devin Smedira[1]

[1]Operations Research Center, MIT, USA, [2]Sloan School of Management, MIT, USA, and [3]Institute for Data Systems and Society, MIT, USA
Corresponding authors: David Gamarnik; Email: gamarnik@mit.edu; Devin Smedira; Email: smedira@mit.edu

(Received 7 October 2025 UTC; revised 28 May 2026 UTC; accepted 27 June 2026 UTC)

Abstract

We design a Correlation-Decay based quasi-polynomial time deterministic approximation algorithm for computing high dimensional integrals, applied in two settings. The first is computing the volume of an independent set polytope with restrictions. Randomized polynomial time approximation algorithms for computing the volume of a convex body have been known now for several decades, but the corresponding deterministic counterparts are not available, and our algorithm is the first of this kind. The class of polytopes for which our algorithm applies arises as linear programming relaxation of the independent set problem with the additional restriction that each variable takes value in the interval $[0, 1-\alpha]$ for some $\alpha < 1/2$. (We note that the $\alpha \geq 1/2$ case is trivial). The method works provided $\alpha > 1/2 - O(1/\Delta^2)$, where Δ is the maximum degree of the graph. When $\Delta = 3$ (the sparsest non-trivial case), our method works provided $0.488 < \alpha < 0.5$. Our second setting is to compute the integral of a multi-dimensional separable function, supported by some underlying hyper-graph structure, appropriately defined. Equivalently, our integral is the partition function of a graphical model with continuous potentials. For our method to work, we require that the domain is bounded and the hyper-edge potentials are positive and bounded on the domain. We further assume that the potentials have upper and lower bounds separated by a multiplicative factor of $1 + O(1/\Delta^2)$, where Δ is the maximum degree of the graph.

Keywords: Correlation decay; spatial mixing; derandomization; approximation algorithms

2020 MSC Codes: Primary: 68W25; Secondary: 60C05, 68Q87

1. Introduction

We consider the problem of designing deterministic polynomial time algorithm for the problem of computing the volume of a polytope. Namely, given a polytope $\mathcal{P}$ described by a sequence of inequalities $\mathcal{P} = \{x : Ax \leq b\}$, where matrix A and vector b are given as problem input, the goal is to compute the volume (Lebesgue measure) of $\mathcal{P}$ using deterministic algorithms with polynomial running time. We further consider an extension of this setup to the problem of designing a deterministic approximation algorithm for computing the integral of a multi-dimensional function over a bounded domain.

The exact computation of polytope volume is known to be #*P*-hard, even in many restrictive setting [13], resulting in a focus on designing approximation algorithms. Fortunately, general approximation algorithms do exist, thanks to the celebrated work by Dyer, Frieze and Kannan [12], who successfully constructed a Markov Chain Monte Carlo (MCMC) based Fully Polynomial

*Support from NSF grants CISE-2233897 and DMS-2015517 is gratefully acknowledged.

© The Author(s), 2026. Published by Cambridge University Press.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

Time Randomized Approximation Scheme (FPRAS) whenever the problem input is a convex body (not necessarily a polytope).

FPRAS is a nearly golden standard in the field of approximation algorithms, with the only caveat being its reliance on randomization. This naturally raises the question of whether a deterministic counterpart exists for the same problem. Broadly, the existence of deterministic counterparts to otherwise randomized algorithms is a central question in theoretical computer science. It is widely believed that any problem which admits polynomial time randomized algorithm (namely belongs to the either of the classes RP or BPP) must admit a deterministic counterpart (namely an algorithm in P) [28]. Unfortunately, in the volume computation case, when the input is a general convex body described via an appropriate oracle, deterministic polynomial approximation algorithms provably do not exist, thanks to a clever argument by Bárány and Füredy [4]. However, their argument does not rule out the existence of deterministic approximation algorithms when the input is an explicitly given polytope, as opposed to an oracle. This question is a central focus of this work.

We have recently witnessed dramatic progress in designing deterministic approximation algorithms for counting problems in graphs. These are problems of the following form: given a graph and a combinatorial structure of a particular type (independent set, matching, proper colouring, etc.), compute approximately the number of such structures in the graph (exact computation is again #P-hard in most cases). A broad generalization of this problem involves computing the object known as the partition function, a canonical object arising in the field of statistical physics.

For a while, the only algorithms for solving these problems was MCMC-based, thus relying on randomization. In 2005, the Correlation Decay method was introduced simultaneously and independently by Bandyobapdhyay and Gamarnik [2, 3] and Weitz [29], which led to deterministic approximation algorithms for counting in graphs. In loose terms, the Correlation Decay property states that when sampling a structure from the Gibbs distribution, the assignment given to vertices which are far apart in the graph are nearly independent. While the algorithm in [2, 3] was restricted to graphs with growing girth, the algorithm in [29], did not have such restrictions and led for the first time to a deterministic Fully Polynomial Time Approximation Scheme (FPTAS) – the strongest algorithm one could hope for this problem. The algorithm was based on an extremely ingenious idea of relating the problem of counting in graphs to the problem of counting in the associated self-avoiding tree, appropriately defined. The exponent of the running time was improved recently to the best possible by developments based on spectral independence [1].

Another method for graph counting problems was introduced by Barvinok [5]. The method relies on rewriting the partition function as a complex polynomial and showing that the logarithm of this polynomial is well approximated by its low order Taylor terms, provided the zeros of the polynomials are outside of a certain complex region containing 0 and 1. Remarkably, the power of the method evidently coincides with the Correlation Decay method, also leading to a deterministic FPTAS in the same regimes. A partial explanation for this was given in [15] and [25].

In contrast, the work on deterministically computing volumes is limited to methods which provide exponential approximations or have runtimes which scale exponentially in the dimension [8, 20]. Since volume computation can be viewed as an idealized version of counting, it is natural to attempt either the Correlation Decay or the interpolation method to address the lack of an FPTAS for this problem. In this work, we choose to adopt the Correlation Decay method. We restrict our attention to polytopes which are linear programming relaxation of the independent set problem and introduce the set up next. Our method may extend far beyond this set up, although serious technical challenges do arise.

Consider a simple undirected graph G on the node set V and edge set E. An independent set is any subset of nodes $I \subset V$ which spans no edges. Namely, for every $u, v \in I$ it is the case that $(u, v) \notin E$. Associate with every node $u \in V$ a variable with range $[0, 1]$ and with every edge (u, v) a linear constraint $x_u + x_v \leq 1$. This gives a rise to a polytope we denote by $\mathcal{P}_{\text{IS}}(G)$. Computing the volume of this polytope algorithmically is our first focus.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

If we consider a further restriction $x_u \in \{0, 1\}$ for every node and count the number of solutions, this clearly corresponds to the number of independent sets in the graph G, which is the subject of [2, 3, 29] and many follow up papers. Indeed, Weitz' algorithm applies to this setting when the largest degree Δ of the graph G does not exceed 5. Importantly, in a dramatic development by Sly [27], it was shown that approximating the number of independent sets in graphs when Δ is allowed to be at least 6 is NP-hard to compute.

At the same time, the approximation algorithm of [12] indicates that such hardness result does not hold for the volume computation problem. Thus, it is likely that a deterministic algorithm exists for this problem. In particular, it is plausible that the Correlation Decay method can be applied in this setting, with a partial progress in this direction having already been achieved in [16]. There, the independent set linear programming relaxation problem described above was considered when the underlying graph is a regular tree. It was established that indeed, consistently with the hypothesis above, the independent set polytope on regular trees exhibits Correlation Decay for *all* degrees. Unfortunately, the method utilized in this paper is not adoptable to graphs with varying degree and new methods are needed. This is what we partially achieve in this paper.

We now describe our first main result. We consider the problem of computing the volume of the polytope $\mathcal{P}_{\mathrm{IS}}(G)$ with an additional assumption that each variable x_v is restricted to be in the interval $[0, 1-\alpha]$ for some $\alpha < 1/2$. We denote this polytope by $\mathcal{P}_{\mathrm{IS}}(G, \alpha)$. We note that the case $\alpha \geq 1/2$ is trivial, as in this case the constraints $x_u + x_v \leq 1$ become redundant and the volume of the polytope is $(1-\alpha)^{|V|}$. The case $\alpha < 1/2$ however is a different story and this is where we achieve a partial progress. We design a deterministic approximation algorithm for this problem with the running time $|V|^{O(\log |V|/\epsilon)}$ for any fixed $\epsilon > 0$. Namely, the value produced by the algorithm normalized by the correct volume is within the interval $(1-\epsilon, 1+\epsilon)$. Thus our algorithm is quasi-polynomial time. We achieve this for the range of Δ-dependent values of α where Δ is the maximum degree of the graph G. Specifically, when $\Delta = 3$, which corresponds to the sparsest graphs for which the problem is non-trivial, our algorithm applies when $\alpha > 0.488$. Asymptotically, our algorithm works when $\alpha > 1/2 - O(1/\Delta^2)$.

We now discuss our algorithm and the rationale for the restriction involving the parameter α. Our algorithm is based on establishing the Correlation Decay property on the appropriately discretized version of the problem. Namely the problem of counting the number of solutions of the system of equations $x_u + x_v \leq N$, $(u, v) \in E$ when x_u are restricted to lie in the set $0, 1, 2, \ldots, N$ for large enough integer value N. An approach for proving the Correlation Decay property, specifically for non-binary models, is to establish a contraction of the functions associated with computation of marginal values, appropriately defined. The contraction in its own turn can be established by showing that $\|\cdot\|_1$ norm of the gradient of this functions is strictly smaller than 1. This method of establishing the Correlation Decay property was introduced in [2, 3] and later re-introduced in [14], and became a fairly canonical approach to similar problems later on (see, for example, [10]). The method introduced by Weitz [29] is based on establishing a certain monotonicity property on the associated self-avoiding tree, (a special case of the computation tree), is evidently limited to 2-valued models, as opposed to the $N+1$-valued setting such as our case.

Unfortunately, the $\|\cdot\|_1$-contraction *does not* hold for the model $\mathcal{P}_{\mathrm{IS}}(G)$, even in the special case when the underlying graph is a regular tree, even though the Correlation Decay property does hold for this model as per the main result in [16] (see Section 3.5 for a numerical verification of this). However, introducing the additional restriction of the form $x_u \in [0, 1-\alpha]$ does induce the $\|\cdot\|_1$ contraction, and this constitutes the main technical result of our paper. The proof of the contraction is somewhat involved, but fairly elementary.

1.1 Extensions to integration

Interested in studying other domains where our technique could be applied, we arrived at the broader question of deterministically approximating generic integrals. The challenge of

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

high-dimensional integration is a central question across many scientific fields, including statistics and machine learning. For example, the question of conducting inference in graphical models, one of the cornerstone statistical models, is reduced to the question of computing the associated partition function, which is a special case of function integration. Exact integration in high dimensions is known to be an intractable problem, as even computing convex volumes is #P-hard. Traditional approaches to approximate integrals also rely on randomized MCMC algorithms, some of which predate the work on volumes. The Metropolis-Hastings algorithm, the development of which dates back to the 1950s [21, 17, 19], is one canonical example of this technique. Subsequent works have introduced several variations on the technique, such as the Metropolis-adjusted Langevin algorithm [26], the Hamiltonian Monte Carlo algorithm [11, 23], and the Metropolis-adjusted Proximal Algorithm [22].

Though we suspect some form of the Correlation Decay property to hold for a wide class of integrable functions, it remains a challenge to define an appropriate notion of the Correlation Decay property for such a general setting, as the notions of 'graph distance' and 'neighbourhoods are not well defined. Thus, we limit ourselves to studying functions $f : \mathbb{R}^n \to \mathbb{R}$ which are defined with respect to some underlying hyper-graph structure. In particular, we require that the function be a product of differentiable *edge potential functions*, each of which depends only on the input values corresponding to the vertices in the edge. Equivalently we consider the graphical model setting with continuous potentials. Under the assumption that the domain of the integration is a rectangle and the minimum and maximum value of the edge potential function over the relevant domain is within a multiplicative factor $1 + O(1/(\eta\Delta))$ of each other, we design a (quasi)-polynomial deterministic approximation algorithm for the integral of f over the associated domain. Here Δ is the maximum number of edges a vertex is a member of, and η is the maximum number neighbours a vertex has in the underlying graph (notice that in the graph setting, $\eta = \Delta$). When $\Delta = 3$ this factor is 1.0479 numerically.

Our algorithm is based on the Correlation Decay property established for a discretized version of the integral problem, equivalent to the classical Riemann Sum. Our approach to prove the Correlation Decay property is to establish that the functions used to evaluate the marginal probabilities exhibit the same style of l_1-contraction as before. We rely on the restriction on the values of the edge potential functions to prove the bound on the gradient norm. We then leverage this to design an algorithm which approximates a Riemann Sum of the function to within a factor of $1 + O(1/|V|)$ in time $|V|^{O(\log |V|)}$ using a similar algorithm as before. Finally, we show that, so long as the edge potential functions have bounded gradients, the value of the Riemann sum well approximates the true value of the integral. This analysis is, in fact, simpler and more streamlined than the polytope case considered earlier.

1.2 Related works and future directions

Admittedly, the regime of applicability of our method is very narrow, in light of the fact that the problem trivializes when $\alpha = 1/2$. Our analysis shows, however, that the problem is very subtle even in this very narrow regime. Furthermore, the subsequent works [7, 9, 18], one of which is devoted to the same problem just introduced, also suffer from a degrading regime of effectiveness as Δ increases, and as a result the authors introduce a similar restriction of the form $\alpha > 1/2 - O(1/\Delta)$, namely just slightly better than ours. Thus it appears that breaking the dependence on Δ presents a substantial barrier.

While we have established the Correlation Decay property for an admittedly rather restricted class of polytopes, we do believe that it holds in full generality. A small step in this direction was achieved in a follow up work, where the interpolation method was used to design a polynomial time algorithm to approximate the volume of the restricted independent set polytope for $\alpha = 1/2 - O(1/\Delta)$ [9]. A further subsequent work applied the same interpolation techniques to a truncated version of the matching polytope, achieving similar bounds [18]. These results highlight

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

the difficulties our current methods have in overcoming the inverse polynomial dependence on degree. Despite this, the existence of a randomized polynomial time approximation algorithms for all convex bodies, as well as the tight connection between the problem of counting in graphs and Correlation Decay, strongly suggests the validity of our Correlation Decay hypothesis. Thus, we pose the following conjecture:

Conjecture 1. *The Gibbs distribution associated with the Lebesgue measure of a polytope exhibits the Correlation Decay property.*

Resolving this conjecture even only for the independent set polytope, or polytopes associated with other combinatorial structures, would itself be a very important step. Our method presented here naturally generalizes to any polytope defined with respect to some bounded-degree hypergraph structure. As such, it is likely that a method to show convergence for any such polytope could be extended to a much more general class of polytopes.

With regards to our integration result, very little is known about deterministic integration in high dimensions. To our knowledge, the only prior work in this direction uses the polynomial interpolation technique to integrate products of quadratic forms with respect to the Gaussian measure in quasi-polynomial time [6]. A subsequent work to ours by the same author considered high-dimensional integration of certain exponential functions using the interpolation technique [7]. Both works require an inverse-polynomial bound on the functions in terms of their 'degree', similar in kind to our $1 + O(1/\Delta)$ bound. Future work inspired by current methods will likely require a similar low degree structure to exploit (see [7] for several such domains of interest).

In contrast to much of the randomized high-dimensional integration literature, such as [22], we do not assume that the function being integrated satisfies assumptions such as log-concavity, log-Sobolev or Poincare inequalities. Therefore, even the existence of randomized algorithms, such as MCMC, is not resolved in our setting. This partially explains why the multiplicative gap $1 + O(1/(\eta\Delta))$ appearing in our set of assumptions is so limited. Thus, we pose this conjecture:

Conjecture 2. *The Glauber dynamics (namely the canonical Markov chain with stationary distribution described by the function we integrate) mixes polynomially fast under the multiplicative gap assumption* $1 + O(1/(\eta\Delta))$.

Similar to computing volumes, we believe that our present assumptions can be significantly relaxed. In general, we paraphrase this as the scientific challenge of establishing deterministic counterparts to randomized integration schemes such as MCMC or Metropolis-Hastings. The $P = RP$ and $P = BPP$ conjectures provide a basis to believe that such deterministic schemes should exist. Furthermore, since the Correlation Decay property is usually tied to isoperimetric properties articulated by log-Sobolev and Poincare type bounds, we suspect it is possible to design such schemes using some form of Correlation Decay.

1.3 Organization of the paper

The remainder of the paper will be divided into two tracks, with one focused on deriving the volume computation results, and the other focused in the integration results. In Section 2, we will separately define each problem, introduce relevant notations, and state the main results. Then, Section 3 will be dedicated to establishing the Correlation Decay property and proving the main result for volume computation. In Subsection 3.5 we will discuss in particular the challenges associated with expanding our result to an unrestricted case $\alpha = 0$. In Section 4 will establish the correlation decay result for integration. Each section (paired with the corresponding introduction in Section 2) will be entirely self contained.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

2. Definitions and main results

In this section we will present the relevant definitions and background information for both the volume computation and integral computation results. We will also overview the Correlation Decay method used to derive the volume computation result at the end of Section 2.1. The second result on integration is derived using a largely similar technique.

When defining these problems, we will reuse notation to mean similar things in each respective context. Only one of the two problems will be discussed per section in all subsequent sections of the paper, so as to prevent ambiguity. Additionally, both problems will be fully defined in this section, despite the repetitions between both setups, so that all notation can be fully defined within each context.

2.1 Volume computation

We now formally restate the definitions of the independent set polytope $\mathcal{P}_{\text{IS}}(G)$, as well as the restricted independent set polytope $\mathcal{P}_{\text{IS}}(G, \alpha)$ from the introduction.

Consider a simple undirected graph $G = (V, E)$ with $m = |V|$ vertices. Let Δ denote the maximum degree of the graph. Namely, letting $d(v) = |\{u : (u, v) \in E\}|$ be the degree of the node v, $\Delta = \max_{v \in V} d(v)$. The independent set polytope $\mathcal{P}_{\text{IS}}(G)$ is defined as

$$\mathcal{P}_{\text{IS}}(G) = \{\mathbf{x} \in \mathbb{R}^m | 0 \leq x_v \leq 1, \forall v \in V, x_u + x_v \leq 1, \forall (u, v) \in E\}.$$

Fix a Restriction Parameter $\alpha \in (0, 1)$. The restricted independent set polytope $\mathcal{P}_{\text{IS}}(G, \alpha)$ is defined as

$$\mathcal{P}_{\text{IS}}(G, \alpha) = \{\mathbf{x} \in \mathbb{R}^m | 0 \leq x_v \leq 1 - \alpha, \forall v \in V, x_u + x_v \leq 1, \forall (u, v) \in E\}.$$

We denote by $\text{Vol}\,(\mathcal{P}_{\text{IS}}(G, \alpha))$ the volume of this polytope. Specifically,

$$\text{Vol}\,(\mathcal{P}_{\text{IS}}(G, \alpha)) = \int 1\,(\mathbf{n} \in \mathcal{P}_{\text{IS}}(G, \alpha))\, d\lambda(\mathbf{n}),$$

where λ denotes the Lebesgue measure in $\mathbb{R}^m$. Our goal is algorithmically computing $\text{Vol}\,(\mathcal{P}_{\text{IS}}(G, \alpha))$ deterministically.

We define $\alpha_\Delta \in (0, 1)$ as the smallest α satisfying

$$\frac{2\Delta(1-\alpha)}{\alpha}\left(\frac{(1-\alpha)^\Delta}{\alpha^\Delta} - 1\right) = 1. \tag{1}$$

Asymptotically, α_Δ is of the order $1/2 - O(1/\Delta^2)$. We now formally state the main result of this work.

Theorem 1 (Approximating the volume of $\mathcal{P}_{\text{IS}}(G, \alpha)$). *There exists a deterministic algorithm which for any graph G, any $\alpha > \alpha_\Delta$, and any $\epsilon > 0$, produces a value $\tilde{V}$ satisfying*

$$1 - \epsilon \leq \frac{\tilde{V}}{\text{Vol}\,(\mathcal{P}_{IS}(G, \alpha))} \leq 1 + \epsilon.$$

The running time of the algorithm is $\left(\frac{m}{\epsilon}\right)^{O(\log(m/\epsilon))}$ time, where $m = |V|$ is the number of nodes in the graph. In particular, the algorithm is a deterministic Fully Quasi-Polynomial Time Approximation Scheme.

We suspect that this result can be improved to obtain a deterministic FPTAS, namely removing the dependence on m and ϵ in the exponent. Several of the original approximation results based on the interpolation method were first of quasi-polynomial time, and later improved to be genuinely polynomial running time algorithms [24].

Next we introduce our main technical result, the Correlation Decay property on the associated computation tree, which is the basis of our algorithm and Theorem 1. Our first step is discretizing

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

the problem and thus converting the problem of volume computation to the problem of counting. Fix a large positive integer N, referred to as the Relaxation Parameter, and define the set

$$\mathcal{D}_{\mathrm{IS}}(G,\alpha,N)=\{\mathbf{x}\in\mathbb{Z}_+^m : x_v\le(1-\alpha)N, \forall v\in V, x_u+x_v\le N, \forall(u,v)\in E\}, \tag{2}$$

and the associated counting measure known as the partition function

$$Z(G)=|\mathcal{D}_{\mathrm{IS}}(G,\alpha,N)|. \tag{3}$$

For linguistic convenience, we will refer to the vectors $\mathbf{x}\in\mathcal{D}_{\mathrm{IS}}(G,\alpha,N)$ as the Relaxed Independent Sets of the graph G. Roughly speaking, as we will make more precise in Section 3.4, the value of $Z(G)$ is the volume of $\mathcal{P}_{\mathrm{IS}}(G,\alpha)$ scaled by N^m. Our main focus is thus developing an approximation algorithm for estimating $Z(G)$.

Due to the recursive nature of the algorithm, we will need to consider the set equation (2) further restricted by a set of equalities of the form $x_v=n$. This type of constraint effectively eliminates one of the m coordinates. A collection of such constraints will be generically denoted by β, and any time an additional constraint of the form $x_v=n$ is added, the collection is updated as $\beta'=\beta\cup(v\leftarrow n)$, which, with some abuse of notation, we write as $(\beta, v\leftarrow n)$. Given a collection β of such constraints we define $\mathcal{D}_{\mathrm{IS}}(G,\alpha,N,\beta)$ as the set of points in the set $\mathcal{D}_{\mathrm{IS}}(G,\alpha,N)$ satisfying the constraints in β. We will denote this set by $\mathcal{D}(G,\beta)$ and refer the its members as the Restricted Relaxed Independent Sets of G. We define by $Z(G,\beta)$ the cardinality of this set.

We now introduce the Gibbs distribution associated with the counting measure $Z(G,\beta)$. This is just a uniform probability measure on the set $\mathcal{D}(G,\beta)$. Namely, we assign probability $1/Z(G,\beta)$ to each point in $\mathcal{D}(G,\beta)$. For each node $v\in V$ and each integer $n\in[0,N]$, we denote by $x(G,v\leftarrow n,\beta)$ the probability that x_v takes value n under this measure. Note that

$$x(G,v\leftarrow n,\beta)=\frac{Z(G,(\beta,v\leftarrow n))}{Z(G,\beta)}.$$

It is possible for both the numerator and the denominator of this expression to be 0. In this case, we define $x(G,v\leftarrow n,\beta)$ to be 0.

A standard approach for computing the partition function $Z(G)$ is to compute marginals $x(G,v\leftarrow n,\beta)$ up to inverse polynomial in m multiplicative errors. This approach goes as follows. Fix any ordering of nodes $v_1,\dots,v_n$ and fix any sequence of values $n_1,\dots,n_m$ in $[0,(1-\alpha)N]$. Specifically we may take all of them equal to zero. Let β_k be the sequence of constraints $v_j\leftarrow 0$, $1\le j\le k$. We have the following telescoping property

$$\begin{aligned}Z(G)&=x^{-1}(G,v_1\leftarrow 0)Z(G,\beta_1)\\&=x^{-1}(G,v_1\leftarrow 0)x^{-1}(G,v_2\leftarrow 0,\beta_1)Z(G,\beta_2)\\&=\prod_{1\le k\le m}x^{-1}(G,v_k\leftarrow 0,\beta_{k-1}),\end{aligned}$$

under the convention that β_0 is an empty set of constraints. Suppose we algorithmically produce a sequence of estimations $y(G,v_k\leftarrow 0,\beta_{k-1})$, $1\le k\le m$ of $x(G,v\leftarrow k,\beta_{k-1})$ up to multiplicative precision say $1/m^2$. Namely,

$$1-1/m^2\le\frac{x(G,v_k\leftarrow 0,\beta_{k-1})}{y(G,v_k\leftarrow 0,\beta_{k-1})}\le 1+1/m^2.$$

Then $\hat{Z}(G)\triangleq\prod_k y^{-1}(G,v_k\leftarrow 0,\beta_{k-1})$ satisfies

$$(1-1/m^2)^m=1-O(1/m)\le\frac{\hat{Z}(G)}{Z(G)}\le(1+1/m^2)^m=1+O(1/m),$$

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

and hence we achieve our goal of estimating $Z(G)$ up to multiplicative approximation better than any fixed constant ϵ for large m. Thus our focus now is designing an algorithm for producing approximations $y(G, v \leftarrow n, \beta)$ of $x(G, v \leftarrow n, \beta)$.

Our next step towards stating the Correlation Decay property is establishing a recursion structure exhibited by the marginal probabilities $x(G, v \leftarrow n, \beta)$. This recursive structure in full generality was introduced in [14] and we adopt it here as well.

We fix a graph $G = (V, E)$, node $v \in V$, and any set of constraints β. Let $v_1, \ldots, v_d$ be the neighbours of v, where $d = d(v)$ is the degree of the node. For any sequence of integers $n_1, \ldots, n_d \in [0, (1-\alpha)N]$, we denote $(n_1, \ldots, n_i)$ by $\bar{n}_i$ for short and consider the associated marginal probabilities

$$x^i(\bar{n}_i) \triangleq x\left(G \setminus v, v_i \leftarrow n_i, \left(\beta, \left(v_j \leftarrow n_j, 1 \le j \le i-1\right)\right)\right).$$

Namely, this is the probability that when nodes $v_1, \ldots, v_{i-1}$ are assigned values $n_1, \ldots, n_{i-1}$ respectively (thus updating the set of constraints from β to $\beta, v_j \leftarrow n_j, j \le i-1$), and v is eliminated from G, the node v_i receives value n_i. The vector $(x^i(\bar{n}_i), 1 \le i \le d)$ of marginals is denoted by $\mathbf{x}$. For every k, we let $S_k = [0, \min\left(N-k, (1-\alpha)N\right)]$. Define the sequence $\left(f_n, n \in [0, (1-\alpha)N]\right)$ as follows:

$$f_n(\mathbf{x}) = \frac{\sum\limits_{n_1, \ldots n_d \in S_n} x^1(\bar{n}_1) \ldots x^d(\bar{n}_d)}{\sum\limits_{m=0}^{(1-\alpha)N} \sum\limits_{n_1, \ldots n_d \in S_m} x^1(\bar{n}_1) \ldots x^d(\bar{n}_d)}. \tag{4}$$

The recursive structure is stated in the following proposition which will be proved in the next section.

Proposition 2. *The following relation holds for every $n \in [0, (1-\alpha)N]$:*

$$x(G, v \leftarrow n, \beta) = f_n(\mathbf{x}).$$

Ultimately, we will use the recursion equation (4) to compute estimations $\mathbf{y}$ of the true marginal probabilities $\mathbf{x}$ recursively. We will introduce the notion of a Feasible Marginal Distribution for this problem setup in Section 3.1.2, and show that both the true marginals $\mathbf{x}$ and the recursive estimates $\mathbf{y}$ satisfy the definition. Our main Correlation Decay result states that provided two vectors $\mathbf{x}$ and $\mathbf{y}$ are Feasible Marginal Distributions and $\alpha < \alpha_\Delta$, where α_Δ satisfies equation (1), the following contraction property holds for every $n \in [0, (1-\alpha)N]$ and for some $0 < c = c(\alpha) < 1$,

$$|f_n(\mathbf{y}) - f_n(\mathbf{x})| \le c\|\mathbf{y} - \mathbf{x}\|_\infty.$$

Namely, every iteration of the recursion encoded in f_n decreases the estimation error geometrically. Provided the number of iterations is of the order $\Theta(\log m)$, we will obtain the required upper $1/m^{\Omega(1)}$ on the estimation error. This is the Correlation Decay on the computation tree property which is the main technical underpinning of our work.

2.2 Integration

We now formally introduce the integration framework presented in the introduction, as well as the corresponding main theorem. Consider a hypergraph $G = (V, E)$ with m vertices of some fixed maximum degree Δ (meaning each vertex in the hypergraph is a member of at most Δ edges). We will further assume that each vertex has at most η neighbours, where a neighbour is defined as any other vertex which shares at least one edge (notice that $\eta \le R\Delta$). We will define a *hypergraph*

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

potential function to be any function $p(\mathbf{n}) : \mathbb{R}^m \to \mathbb{R}$ which takes the form

$$p(\mathbf{n}) = \prod_{e \in E} \Phi_e(\mathbf{n}_e). \tag{5}$$

We refer to each function Φ_e as the edge potential function for the edge e. We wish to compute the expression

$$\int 1(\mathbf{n} \in [0,1]^m) p(\mathbf{n}) d\lambda(\mathbf{n}),$$

where λ is the Lebesgue measure. Though we define the region of integration to be $[0,1]^m$ here for convenience, similar results will hold for any rectangular domain. Rather than computing a direct approximation of this integral, we will instead compute an approximation of the corresponding Riemann Sum. Fix some relaxation parameter N, and let $\mathcal{D} = \{\mathbf{n}/N | \mathbf{n} \in \mathbb{Z}^m, 0 \leq n_v < N \; \forall v \in V\}$. We define the partition function

$$Z(G) = \sum_{\mathbf{n} \in \mathcal{D}} p(\mathbf{n}), \tag{6}$$

which is equivalent to a traditional Riemann Sum scaled up by a factor of N^m, with the goal of approximating this value. We will restrict ourselves to the setting where each Φ_e above satisfies

$$c_e \leq \Phi_e(\mathbf{n}_e) \leq (1-\delta)^{1/2\Delta} c_e \left(1 + \frac{\log(1+\frac{1}{\eta})}{2\Delta}\right), \tag{7}$$

for some constant $c_e > 0$ unique to each edge and a universal constant $\delta > 0$ which controls the rate of convergence for our algorithm. In this setting, we will be able to show that the correlation decay property holds, once more allowing us to approximate the value $Z(G)$. We will then be able to relate this approximation back to the value of the original integral. In particular, we will show the following.

Theorem 3 (Approximating integrals of hypergraph potential functions). *Suppose we have a function p taking the form of equation (5) with respect to some hypergraph G. Further, suppose each function Φ_e is differentiable, satisfies the condition in equation (7), and there exists some $k > 0$ such that every Φ_e satisfies $\|\nabla \Phi_e(\mathbf{n}_e)\|_2 \leq k c_e$ for every $\mathbf{n} \in [0,1]^m$. Then, there exists an algorithm to produce an estimate $\tilde{V}$ of $\int 1(\mathbf{n} \in [0,1]^m) p(\mathbf{n}) d\lambda$ such that*

$$\left| \frac{\tilde{V}}{\int 1(\mathbf{n} \in [0,1]^m) p(\mathbf{n}) d\lambda} - 1 \right| \leq O(1/m^2).$$

Taking k, Δ, and η to be constants, this algorithm will run in $m^{O(\log(m))}$ time.

In the course of our approximation algorithm, we will again need to iteratively restrict the domain of our partition function. Thus, we will introduce constraints of the form $v \leftarrow n$, equivalent to restricting $\mathbf{n}_v = n$, and let β be collections of such constraints. We then define $\mathcal{D}_\beta \subset \mathcal{D}$ to be the set of points in $\mathcal{D}$ that satisfy every constraint in β. We will use the notation $(\beta, v \leftarrow n)$ to denote $\beta \cup \{v \leftarrow n\}$. With this notation, we can define the following family of restricted partition functions

$$Z(G, \beta) = \sum_{\mathbf{n} \in \mathcal{D}_\beta} p(\mathbf{n}). \tag{8}$$

We again use this to define a probability distribution over the assignments to each vertex in the hypergraph G induced by the function f and constraints β. In particular, we define

$$x(G, v \leftarrow n, \beta) = \frac{Z(G, (\beta, v \leftarrow n))}{Z(G, \beta)},$$

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

which for each well formulated β will define a probability distribution over the assignments 0 to $N-1$ for the vertex v. Our algorithm will rely on computing an approximation $\tilde{x}(G, v \leftarrow n, \beta)$ of the above value, which will be done by exploiting the recursive nature of the problem. In particular, if we order the vertices 1 through m and let $\beta_i = (v_j \leftarrow 0, j \leq i)$ for $i \in [0, m]$, with $\beta_0 = \{\}$, we can see

$$\begin{aligned} Z(G) &= x^{-1}(G, v_1 \leftarrow 0, \beta_0) Z(G, \beta_1) \\ &= \prod_{i=1}^{m} x^{-1}(G, v_i \leftarrow 0, \beta_{i-1}) Z(G, \beta_m) \\ &= p(\mathbf{0}) \prod_{i=1}^{m} x^{-1}(G, v_i \leftarrow 0, \beta_{i-1}), \end{aligned} \tag{9}$$

since $Z(G, \beta_m) = p(\mathbf{0})$ by construction. Thus, if we can approximate the values of $x(G, v_i \leftarrow n, \beta_{i-1})$ within a factor of $\frac{1}{m^2}$, we will be able to approximate $Z(G)$ within a factor of $\frac{1}{m}$. We will accomplish this with the same high level technique used in the polytope section.

3. Proof of Theorem 1

This section will be dedicated to building up the proof of Theorem 1. Section 3.1 will establish Proposition 2, as well as describe some basic properties of $f_n(\mathbf{x})$. Section 3.2 will then build on these basic properties to prove our main contraction result Theorem 16, a bound on the norm of ∇f. We then formally present our algorithm for approximating the partition function $Z(G)$ in Section 3.3, with Section 3.4 finishing our proof by connecting this back to estimating the volume of $\mathcal{P}_{\text{IS}}(G, \alpha)$. Finally, we discuss some barriers to solving the $\alpha = 0$ in Section 3.5.

3.1 Preliminary results

This section will be primarily focused on providing a complete proof of Proposition 2. We will then provide a definition of a Feasible Marginal Distribution and establish a few related properties.

3.1.1 Proof of Proposition 2

We will now show how to write an expression for $x(G, v \leftarrow n, \beta)$ in terms of the variables $x^i(\bar{n}_i)$, which represent the probability distribution over the assignments of adjacent vertices in smaller graphs. We will again assume that the vertex v has $d \leq \Delta$ neighbours $v_1, \ldots v_d$. We start this process by proving the next two Lemmas.

Lemma 4 (Recursive deconstruction).

$$x(G, v \leftarrow n, \beta) = x(G, v \leftarrow 0, \beta) \sum_{n_1, \ldots n_d \in S_n} x^1(\bar{n}_1) \ldots x^d(\bar{n}_d)$$

Proof. Notice that

$$Z(G, (\beta, v \leftarrow 0)) = Z(G \setminus v, \beta),$$

and more generally

$$Z(G, (\beta, v \leftarrow n)) = \sum_{(n_1, \ldots n_d) \in S_n^d} Z(G \setminus v, (\beta, (v_i \leftarrow n_i \forall i))).$$

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

Using this, we have

$$
\begin{aligned}
&x(G, v \leftarrow n, \beta) \\
&\quad = \frac{Z(G, (\beta, v \leftarrow n))}{Z(G, \beta)} \\
&\quad = \frac{Z(G, (\beta, v \leftarrow 0))Z(G, (\beta, v \leftarrow n))}{Z(G, \beta)Z(G \setminus v, \beta)} \\
&\quad = \frac{x(G, v \leftarrow 0, \beta)}{Z(G \setminus v, \beta)} Z(G, (\beta, v \leftarrow n)) \\
&\quad = \frac{x(G, v \leftarrow 0, \beta)}{Z(G \setminus v, \beta)} \sum_{(n_1, \ldots n_d) \in S_n^d} Z(G \setminus v, (\beta, (v_i \leftarrow n_i \forall i))) \\
&\quad = \frac{x(G, v \leftarrow 0, \beta)}{Z(G \setminus v, \beta)} \sum_{n_1 \in S_n} Z(G \setminus v, (\beta, v_1 \leftarrow n_1)) \sum_{(n_2, \ldots n_d) \in S_n^{d-1}} \frac{Z(G \setminus v, (\beta, (v_i \leftarrow n_i \forall i)))}{Z(G \setminus v, (\beta, v_1 \leftarrow n_1))} \\
&\quad = x(G, v \leftarrow 0, \beta) \sum_{n_1 \in S_n} \frac{Z(G \setminus v, (\beta, v_1 \leftarrow n_1))}{Z(G \setminus v, \beta)} \sum_{(n_2, \ldots n_d) \in S_n^{d-1}} \frac{Z(G \setminus v, (\beta, (v_i \leftarrow n_i \forall i)))}{Z(G \setminus v, (\beta, v_1 \leftarrow n_1))} \\
&\quad = x(G, v \leftarrow 0, \beta) \sum_{n_1 \in S_n} x(G \setminus v, v_1 \leftarrow n_1, \beta) \sum_{(n_2, \ldots n_d) \in S_n^{d-1}} \frac{Z(G \setminus v, (\beta, (v_i \leftarrow n_i \forall i)))}{Z(G \setminus v, (\beta, v_1 \leftarrow n_1))} \\
&\quad = x(G, v \leftarrow 0, \beta) \sum_{n_1 \in S_n} x^1(\bar{n}_1) \sum_{(n_2, \ldots n_d) \in S_n^{d-1}} \frac{Z(G \setminus v, (\beta, (v_i \leftarrow n_i \forall i)))}{Z(G \setminus v, (\beta, v_1 \leftarrow n_1))} \\
&\quad = x(G, v \leftarrow 0, \beta) \sum_{n_1 \in S_n} x^1(\bar{n}_1) \ldots \sum_{n_d \in S_n} x^d(\bar{n}_d) \\
&\quad = x(G, v \leftarrow 0, \beta) \sum_{n_1, \ldots n_d \in S_n} x^1(\bar{n}_1) \ldots x^d(\bar{n}_d),
\end{aligned}
$$

where the second to last line is derived by repeating the same logic for each neighbour v_i with a, appropriately modified value of β. □

Lemma 5 (Normalizing term).

$$
x(G, v \leftarrow 0, \beta) = \frac{1}{\sum_{m=0}^{(1-\alpha)N} \sum_{n_1, \ldots n_d \in S_m} x^1(\bar{n}_1) \ldots x^d(\bar{n}_d)}.
$$

Proof. By construction, $\sum_{m=0}^{(1-\alpha)N} x(G, v \leftarrow m, \beta) = 1$, since $((x(G, v \leftarrow m, \beta))$ is a probability vector for $m \in [0, (1-\alpha)N]$. Thus, by Lemma 4,

$$
1 = \sum_{m=0}^{(1-\alpha)N} x(G, v \leftarrow m, \beta) = x(G, v \leftarrow 0, \beta) \sum_{m=0}^{(1-\alpha)N} \sum_{n_1, \ldots n_d \in S_m} x^1(\bar{n}_1) \ldots x^d(\bar{n}_d).
$$

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

Rearranging terms, we have

$$x(G, v \leftarrow 0, \beta) = \frac{1}{\sum_{m=0}^{(1-\alpha)N} \sum_{n_1,\ldots n_d \in S_m} x^1(\bar{n}_1) \ldots x^d(\bar{n}_d)}.$$

□

Combining Lemmas 4 and 5, we immediately obtain Proposition 2.

3.1.2 Bounds on marginal probabilities

Section 3.2 will be dedicated to establishing a bound on the 1-norm of the function f_n from equation (4). The bound we compute will need to hold not just on the true probability distribution $\mathbf{x}$, but also on any linear interpolation between the true distribution and our probability estimates. To do this, we identify the primary properties we need for the contraction to hold. Thus, we introduce the following notion of a Feasible Marginal Distribution, and show that our bound will hold on all such distributions.

Definition 1 (Feasible marginal distribution). *Fix some* $\mathbf{w} = \{w_k > 0, 0 \leq k \leq (1-\alpha)N\}$. *We say that* $\mathbf{w}$ *is a* Feasible Probability Distribution *if*

$$\sum_{k=0}^{(1-\alpha)N} w_k = 1, \tag{10}$$

$$w_k \geq w_{k+1}, \tag{11}$$

$$w_k = w_{k'} \qquad \forall k, k' \leq \alpha N. \tag{12}$$

Further, we say $\mathbf{y} = \{y^j(\bar{n}_j) | j \in \{1, \ldots d\}, \bar{n}_j \in [0, (1-\alpha)N]^j\}$ *is a* Feasible Marginal Distribution *if for every* $j \in \{1, \ldots d\}$ *and* $\bar{n}_{j-1} \in [0, (1-\alpha)N]^{j-1}$ *where* $(\beta, v_i \leftarrow n_i, i < j)$ *forms a feasible set of constraints, the values* $y_i = y^j(\bar{n}_{j-1}, i)$ *for* $i \in [0, (1-\alpha)N]$ *form a Feasible Probability Distribution and for every* $\bar{n}_j, \bar{m}_j \in [0, \alpha N]$,

$$y^j(\bar{n}_j) = y^j(\bar{m}_j). \tag{13}$$

We now show that $\mathbf{x}$ as defined above is a Feasible Marginal Distribution. Firstly, observe from the problem's construction that for any fixed $j, n_1, \ldots n_{j-1}$, the set $\{x^j(\bar{n}_{j-1}, n_j) | n_j \in [0, (1-\alpha)N]\}$ forms a probability distribution when $(\beta, v_i \leftarrow n_i, i < j)$ forms a feasible set of constraints, meaning equation (10) is satisfied. Next, notice that $Z(G, (\beta, v \leftarrow n)) \geq Z(G, (\beta, v \leftarrow n+1))$, since for every relaxed independent set where v is assigned $n+1$ there is a relaxed in dependent that is identical except v is assigned n. Thus, we have that for every fixed $\bar{n}_{j-1}, x^j(\bar{n}_{j-1}, n_j) \geq x^j(\bar{n}_{j-1}, n_j + 1)$, so $\mathbf{x}$ satisfies equation (11). Finally, we have that $Z(G, (\beta, v \leftarrow n)) = Z(G, (\beta, v \leftarrow n'))$ for every $0 \leq n, n' \leq \alpha N$, since assigning any vertex a value in $[0, \alpha N]$ cannot violate any constraints. More generally, if every constraint in β and β' assigns a value of at most αN, then $Z(G, \beta) = Z(G, \beta')$. This implies $\mathbf{x}$ must satisfy equations (12) and (13). Therefore, $\mathbf{x}$ is a Feasible Marginal Distribution.

Lemma 6 (Closure of feasible marginal distributions under f_n). *Let* $\mathbf{y}$ *be a Feasible Marginal Distribution, and let* $y'_n = f_n(\mathbf{y})$ *for* $n \in [0, (1-\alpha)N]$. *Then, the set of* y'_n *form a Feasible Probability Distribution.*

Proof. Notice that $S_n = S_{n'}$ for any $n, n' \leq \alpha N$ by construction. By equation (4), we can clearly see this implies $f_n(\mathbf{y}) = f_{n'}(\mathbf{y})$ when $n, n' \leq \alpha N$, so the values of y'_n satisfy equation (12). Equation (10) is satisfied because

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

$$\sum_{n=0}^{(1-\alpha)N} f_n(\mathbf{y}) = \frac{\sum_{n=0}^{(1-\alpha)N} \sum_{n_1,\dots n_d \in S_n} y^1(\bar{n}_1)\dots y^d(\bar{n}_d)}{\sum_{m=0}^{(1-\alpha)N} \sum_{n_1,\dots n_d \in S_m} y^1(\bar{n}_1)\dots y^d(\bar{n}_d)} = 1.$$

Finally, equation (11) is clearly satisfied because S_n is non-increasing in n and each term in the numerator sum is non-negative. □

Lemma 7 (Linear interpolations of feasible Marginal distributions). *Let* $\mathbf{y}$ *and* $\mathbf{z}$ *be two Feasible Marginal Distributions. Then, for any* $w \in [0,1]$, $w\mathbf{y} + (1-w)\mathbf{z}$ *is a Feasible Marginal Distribution, where the sum is taken pointwise.*

The proof of Lemma 7 follows directly from the linearity of the four equations (10), (11), (12), and (13). These results will allow us to construct the contraction in sufficient generality. We conclude by proving two Lemmas about Feasible Probability Distributions.

Lemma 8. *For any Feasible Probability Distribution* $\mathbf{y}$*, we have*

$$y_i \leq \frac{1}{\alpha N}.$$

Proof. By equation (12), for any $i, j \in [0, \alpha N]$, the value of $y_i = y_j$. There are αN such values, so each one must be no greater than $\frac{1}{\alpha N}$. Additionally, for any i, $y_i \leq y_0 \leq \frac{1}{\alpha N}$ by equation (11), finishing the proof. □

Lemma 9. *For any Feasible Probability Distribution* $\mathbf{y}$*, we have*

$$\frac{\alpha}{1-\alpha} \leq \sum_{i=0}^{\alpha N} y_i \leq 1.$$

Proof. The upper bound holds directly from equation (10). To prove the lower bound, first observe equation (10) implies $\sum_{i=0}^{(1-\alpha)N} y_i = 1$. If $\sum_{i=0}^{\alpha N} y_i < \frac{\alpha}{1-\alpha}$, then it would have to be the case that $\sum_{i=\alpha N+1}^{(1-\alpha)N} y_i > \frac{1-2\alpha}{1-\alpha}$. This would imply the last $1-2\alpha$ terms in the sequence had a larger average value than the first α terms. Equation (11) implies this sequence must be decreasing, which is a contradiction. Thus, the bound must hold. □

3.2 Proof of the polytope correlation decay property

In this section, we establish the key correlation decay property necessary to design the approximation algorithm. The primary result, occurring at the end of the section, will be a bound on $\|f_n(\mathbf{y})\|_1$ defined in equation (4) with respect to a vertex v with d neighbours for any Feasible Marginal Distribution $\mathbf{y}$. The bound will be in terms of α and d, and will be bounded away from 1 as α approaches $1/2$ for any fixed d.

3.2.1 Expressions for partial derivatives

We now introduce the following two expressions to represent the numerator and denominator of equation (4):

$$U_n = \sum_{n_1,\dots n_d \in S_n} y^1(\bar{n}_1)\dots y^d(\bar{n}_d),$$

$$L = \sum_{m=0}^{(1-\alpha)N} \sum_{n_1,\dots n_d \in S_m} y^1(\bar{n}_1)\dots y^d(\bar{n}_d) = \sum_{m=0}^{(1-\alpha)N} U_m.$$

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

In particular, we have that $f_n(\mathbf{y}) = \frac{U_n}{L}$. By the chain rule, we can see

$$\frac{\partial f_n(\mathbf{y})}{\partial y^j(\bar{n}_j)} = \frac{L\frac{\partial U_n(\mathbf{x})}{\partial x^j(\bar{n}_j)} - U_n\frac{\partial L(\mathbf{x})}{\partial x^j(\bar{n}_j)}}{L^2}. \tag{14}$$

Let

$$B_j(\bar{n}_{j-1}) = y^1(\bar{n}_1)\dots y^{j-1}(\bar{n}_{j-1}),$$
$$A_j(n,\bar{n}_j) = \sum_{n_{j+1},\dots n_d \in S_n} y^{j+1}(\bar{n}_{j+1})\dots y^d(\bar{n}_d).$$

Thus, for any $j \in [1,d]$, we have

$$U_n = \sum_{n_1,\dots n_{j-1}\in S_n} B_j(\bar{n}_{j-1}) \sum_{n_j\in S_n} y^j(\bar{n}_j)A_j(n,\bar{n}_j), \tag{15}$$

$$L = \sum_{m=0}^{(1-\alpha)N} \sum_{n_1,\dots n_{j-1}\in S_m} B_j(\bar{n}_{j-1}) \sum_{n_j\in S_m} y^j(\bar{n}_j)A_j(m,\bar{n}_j). \tag{16}$$

Fix $j \in [1,d]$ and some $\bar{n}_j \in [0,(1-\alpha)N]^j$. Letting $I(A)$ be the indicator variable for the event A, we obtain

$$\frac{\partial U_n}{\partial x^j(\bar{n}_j)} = I(\bar{n}_j \in S_n^d)B_j(\bar{n}_{j-1})A_j(n,\bar{n}_j), \tag{17}$$

$$\frac{\partial L}{\partial x^j(\bar{n}_j)} = \sum_{m=0}^{(1-\alpha)N} I(\bar{n}_j \in S_m^d)B_j(\bar{n}_{j-1})A_j(m,\bar{n}_j). \tag{18}$$

Lemma 10. *For any $j \in [1,d]$, $n \in [0,(1-\alpha)N]$, and $n_1,\dots n_j \in [0,(1-\alpha)N]$, we have*

$$A_j(n,\bar{n}_j) \le 1,$$
$$B_j(\bar{n}_{j-1}) \le \frac{1}{(\alpha N)^{j-1}}.$$

Proof. Recall that for any $\bar{n}_{d-1}$, $y^d(\bar{n}_d)$ is a probability distribution as n_d varies. Thus,

$$\sum_{n_d\in S_n} y^d(\bar{n}_d) \le \sum_{n_d=0}^{(1-\alpha)N} y^d(\bar{n}_d) = 1.$$

Iterating this process for vertices $v_d, v_{d-1},\dots v_{j+1}$ leads to the bound on A_j. The bound on B_j follows from Lemma 8. □

Lemma 11. *For any $j \in [1,d]$, $n \in [0,(1-\alpha)N]$, $n_1,\dots n_j \in [0,\alpha N]$, and $m_1,\dots m_j \in [0,\alpha N]$ we have that*

$$A_j(n,\bar{n}_j) = A_j(n,\bar{m}_j)$$
$$B_j(\bar{n}_{j-1}) = B_j(\bar{m}_{j-1})$$

Proof. By definition, we have $y^{j'}(\bar{n}_{j'}) = y^{j'}(\bar{m}_{j'})$ for any $j' \le j$. Repeatedly substituting these equalities into the definitions of A_j and B_j produces the desired result. □

We now move to bounding our numerator and denominator functions, U_n and L, directly.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

Lemma 12 (Gradient denominator).

$$L \geq \alpha N$$

Proof. Fix any $m \leq \alpha N$. We will have that $S_m = [0, \min(N-m, (1-\alpha)N] = [0, (1-\alpha)N]$. For any $n_1, \ldots n_{j-1}$, $\sum_{n_j=0}^{\alpha N} y^j(\bar{n}_j) = 1$ since $\mathbf{y}$ is a Feasible Marginal Distribution. Thus,

$$\sum_{n_1,\ldots n_d \in S_m} y^1(\bar{n}_1)\ldots y^d(\bar{n}_d) = \sum_{n_1,\ldots n_d=0}^{(1-\alpha)N} y^1(\bar{n}_1)\ldots y^d(\bar{n}_d) = 1.$$

We then obtain

$$L = \sum_{m=0}^{(1-\alpha)N} \sum_{n_1,\ldots n_d \in S_m} y^1(\bar{n}_1)\ldots y^d(\bar{n}_d) \geq \sum_{m=0}^{\alpha N} \sum_{n_1,\ldots n_d \in S_m} y^1(\bar{n}_1)\ldots y^d(\bar{n}_d) = \sum_{m=0}^{\alpha N} 1 = \alpha N.$$

□

Lemma 13 (Upper bounds). *For any $j \in [1,d]$, $n \in [0,(1-\alpha)N]$, and $n_1, \ldots n_{j-1} \in [0,(1-\alpha)N]$, we have*

$$\begin{aligned} U_n &\leq 1, \\ L &\leq (1-\alpha)N, \\ \frac{\partial U_n(\boldsymbol{x})}{\partial x^j(\bar{n}_j)} &\leq \frac{1}{(\alpha N)^{j-1}}, \\ \frac{\partial L(\boldsymbol{x})}{\partial x^j(\bar{n}_j)} &\leq \frac{(1-\alpha)N}{(\alpha N)^{j-1}}. \end{aligned}$$

Proof. We will show the bounds on U_n and $\frac{\partial U_n(\mathbf{x})}{\partial x^j(\bar{n}_j)}$. These will immediately imply the bounds on L and $\frac{\partial L(\mathbf{x})}{\partial x^j(\bar{n}_j)}$ by definition. Writing U_n in terms of A_1 in equation (15), we see that

$$U_n = \sum_{n_1 \in S_n} y^j(\bar{n}_1) A_1(n, \bar{n}_1) \leq \sum_{n_1 \in S_n} y^j(\bar{n}_1) \leq 1,$$

where the first inequality follows from Lemma 10 and the second follows from equation (10). Similarly, we can write $\frac{\partial U_n(\mathbf{x})}{\partial x^j(\bar{n}_j)}$ in terms of A_j and B_j as in equation (17) to see that

$$\frac{\partial U_n(\mathbf{x})}{\partial x^j(\bar{n}_j)} = I(\bar{n}_j \in S_n^d) B_j(\bar{n}_{j-1}) A_j(n, \bar{n}_j) \leq B_j(\bar{n}_{j-1}) A_j(n, \bar{n}_j) \leq \frac{1}{(\alpha N)^j},$$

where the last inequality follows from Lemma 10. □

3.2.2 Upper bounds on partial derivatives

We are now obtain bounds on the value of the partial derivatives of $f(\mathbf{y})$, in order to ultimately bound $\|\nabla f(\mathbf{y})\|_1$. We will do this by using two different bounds on $\frac{\partial f(\mathbf{y})}{\partial y^j(\bar{n}_j)}$, depending on the value of $\bar{n}_j$. We present the looser bound first, which applies to all partial derivatives.

Theorem 14 (Loose partial bound). *For any $j \in [1,d]$ and $\bar{n}_j \in [0,(1-\alpha)N]^j$,*

$$\left| \frac{\partial f(\mathbf{y})}{\partial y^j(\bar{n}_j)} \right| \leq \frac{(1-\alpha)N}{(\alpha N)^{j+1}}$$

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

Proof. As we saw in equation (14), we have

$$\begin{aligned}\left|\frac{\partial f(\mathbf{y})}{\partial y^j(\bar{n}_j)}\right| &= \left|\frac{L\frac{\partial U_n(\mathbf{x})}{\partial x^j(\bar{n}_j)} - U_n\frac{\partial L(\mathbf{x})}{\partial x^j(\bar{n}_j)}}{L^2}\right| \\ &\leq \frac{\max\left(L\frac{\partial U_n(\mathbf{x})}{\partial x^j(\bar{n}_j)}, U_n\frac{\partial L(\mathbf{x})}{\partial x^j(\bar{n}_j)}\right.}{L^2} \\ &\overset{\text{a}}{\leq} \frac{\max\left(L\frac{\partial U_n(\mathbf{x})}{\partial x^j(\bar{n}_j)}, U_n\frac{\partial L(\mathbf{x})}{\partial x^j(\bar{n}_j)}\right)}{(\alpha N)^2} \\ &\overset{\text{b}}{\leq} \frac{(1-\alpha)N}{(\alpha N)^{j+1}},\end{aligned}$$

where inequality a follows from Lemma 12 and inequality b follows from Lemma 13. □

As we will see in the next section, this bound alone will not be sufficiently tight. Next, we introduce a tighter bound on the partial derivative for cases where $\bar{n}_j \in [0, \alpha N]^j$.

Theorem 15 (Tighter partial bound). *For any $j \in [1, d]$ and $\bar{n}_j \in [0, \alpha N]^j$,*

$$\left|\frac{\partial f(\mathbf{y})}{\partial y_j(\bar{n}_j)}\right| \leq \frac{j(1-2\alpha)(1-\alpha)^j}{\alpha^{2j+1}N^j}.$$

Proof.

$$\begin{aligned}&\left|\frac{\partial f(\mathbf{y})}{\partial y_j(\bar{n}_j)}\right| \\ &= \frac{|L\frac{\partial U_n(\mathbf{x})}{\partial x^j(\bar{n}_j)} - U_n\frac{\partial L(\mathbf{x})}{\partial x^j(\bar{n}_j)}|}{L^2} \\ &\overset{\text{a}}{\leq} \frac{1}{(\alpha N)^2}|L\frac{\partial U_n(\mathbf{x})}{\partial x^j(\bar{n}_j)} - U_n\frac{\partial L(\mathbf{x})}{\partial x^j(\bar{n}_j)}| \\ &\overset{\text{b}}{=} \frac{1}{(\alpha N)^2}\left|\sum_{m=0}^{(1-\alpha)N}\left[\sum_{\bar{m}_j\in S_m^j}\frac{\partial U_n(\mathbf{x})}{\partial x^j(\bar{n}_j)}B_j(\bar{m}_{j-1})y^j(\bar{m}_j)A_j(m,\bar{m}_j) - U_n I(\bar{n}_j\in S_m^j)B_j(\bar{n}_{j-1})A_j(m,\bar{n}_j)\right]\right| \\ &\overset{\text{c}}{=} \frac{1}{(\alpha N)^2}\left|\sum_{m=0}^{(1-\alpha)N}\left[\sum_{\bar{m}_j\in S_m^j} I(\bar{n}_j\in S_n^j)B_j(\bar{n}_{j-1})A_j(n,\bar{n}_j)B_j(\bar{m}_{j-1})y^j(\bar{m}_j)A_j(m,\bar{m}_j)\right.\right. \\ &\quad \left.\left. - \sum_{\bar{p}_j\in S_n^j} B_j(\bar{p}_{j-1})y^j(\bar{p}_j)A_j(n,\bar{p}_j)I(\bar{n}_j\in S_m^j)B_j(\bar{n}_{j-1})A_j(m,\bar{n}_j)\right]\right| \\ &\overset{\text{d}}{=} \frac{B_j(\bar{n}_{j-1})}{(\alpha N)^2}\left|\sum_{m=0}^{(1-\alpha)N}\left[\sum_{\bar{m}_j\in S_m^j}\left\{A_j(n,\bar{n}_j)B_j(\bar{m}_{j-1})y^j(\bar{m}_j)A_j(m,\bar{m}_j)\right\}\right.\right. \\ &\quad \left.\left. - \sum_{\bar{p}_j\in S_n^j}\left(B_j(\bar{p}_{j-1})y^j(\bar{p}_j)A_j(n,\bar{p}_j)A_j(m,\bar{n}_j)\right)\right]\right|\end{aligned}$$

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

where inequality a comes from Lemma 12 , and equalities b and c come from applying equations (16), (18), (15), and (17). Equality d comes from factoring out $B_j(\bar{n}_{j-1})$ and observing that $\bar{n}_j \in S_n^j, S_m^j$ because we assumed it was in $[0, \alpha N]^j$, which is a subset of S_k for every k.

We observe that when $\bar{m}_j, \bar{p}_j \in [0, \alpha N]^j$, the expression above in the braces {} will be equal to the expression in the parentheses () by Lemma 11 and equation (13). Additionally, we bound the expression in the braces {}. Using Lemmas 8 and 10, we have that for any $\bar{m}_j$

$$A_j(n, \bar{n}_j)B_j(\bar{m}_{j-1})y^j(\bar{m}_j)A_j(m, \bar{m}_j) \leq \frac{1}{(\alpha N)^j},$$

with the same bound being true for the expression in the parentheses (). Thus, we see

$$\begin{aligned}
\left|\frac{\partial f(\mathbf{y})}{\partial y_j(\bar{n}_j)}\right| &\leq \frac{B_j(\bar{n}_{j-1})}{(\alpha N)^2} \sum_{m=0}^{(1-\alpha)N} \left[\max \left(\sum_{\bar{m}_j \in S_m^j, \bar{m}_j \notin [0,\alpha N]^j} A_j(n, \bar{n}_j)B_j(\bar{m}_{j-1})y^j(\bar{m}_j)A_j(m, \bar{m}_j), \right.\right. \\
&\qquad \left.\left. \sum_{\bar{p}_j \in S_n^j, \bar{p}_j \notin [0,\alpha N]^j} B_j(\bar{p}_{j-1})y^j(\bar{p}_j)A_j(n, \bar{p}_j)A_j(m, \bar{n}_j) \right) \right] \\
&\overset{\text{a}}{\leq} \frac{B_j(\bar{n}_{j-1})}{(\alpha N)^2} \sum_{m=0}^{(1-\alpha)N} \frac{N^j[(1-\alpha)^j - \alpha^j]}{(\alpha N)^j} \\
&\overset{\text{b}}{\leq} \frac{1}{(\alpha N)^{j+1}} \sum_{m=0}^{(1-\alpha)N} \frac{N^j[(1-\alpha)^j - \alpha^j]}{(\alpha N)^j} \\
&= \frac{(1-\alpha)N}{(\alpha N)^{j+1}} \left(\frac{(1-\alpha)^j}{\alpha^j} - 1 \right)
\end{aligned}$$

Inequality b comes from Lemma 10. Inequality a comes from the bound above, and observing that there are at most $N^j(1-\alpha)^j$ vectors in S_m^j or S_n^j and exactly $N^j\alpha^j$ vectors in $[0, \alpha N]^j$. □

3.2.3 Bounding the gradient

We are now able to prove a bound on the gradient of the function in equation (4) in terms of d and α.

Theorem 16 (Bounded gradient). *For any $n \in [0, (1-\alpha)N]$,*

$$\|\nabla f_n(\mathbf{y})\|_1 \leq \frac{2d(1-\alpha)}{\alpha} \left(\frac{(1-\alpha)^d}{\alpha^d} - 1 \right).$$

In particular, when G has maximum degree Δ, $\|\nabla f_n(\mathbf{y})\|_1 < 1$ so long as $\alpha > \alpha_\Delta$, where α_Δ is defined in equation (1).

Proof. First, fix some $j \in [1, d]$. Notice that there are exactly $(\alpha N)^j$ different $\bar{n}_j \in [0, \alpha N]^j$. Further, there are $(1-\alpha)^j N^j$ different $\bar{n}_j \in [0, (1-\alpha)N]^j$, meaning there are $N^j[(1-\alpha)^j - \alpha^j]$ vectors in the latter not in the former.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use , available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

Using theorems 14 and 15, we obtain

$$\sum_{\bar{n}_j \in [0,(1-\alpha)N]^j} \left| \frac{\partial f(\mathbf{y})}{\partial y^j(\bar{n}_j)} \right| \leq (\alpha N)^j \frac{(1-\alpha)N}{(\alpha N)^{j+1}} \left(\frac{(1-\alpha)^j}{\alpha^j} - 1 \right) + N^j((1-\alpha)^j - \alpha^j) \frac{(1-\alpha)N}{(\alpha N)^{j+1}}$$

$$= \frac{2(1-\alpha)}{\alpha} \left(\frac{(1-\alpha)^j}{\alpha^j} - 1 \right).$$

Using this, we can conclude

$$\begin{aligned} \|\nabla f_n(\mathbf{y})\|_1 &= \sum_{j=1}^{d} \sum_{\bar{n}_j \in [0,(1-\alpha)N]^j} \left| \frac{\partial f(\mathbf{y})}{\partial y^j(\bar{n}_j)} \right| \\ &\leq \sum_{j=1}^{d} \frac{2(1-\alpha)}{\alpha} \left(\frac{(1-\alpha)^j}{\alpha^j} - 1 \right) \\ &\leq \sum_{j=1}^{d} \frac{2(1-\alpha)}{\alpha} \left(\frac{(1-\alpha)^d}{\alpha^d} - 1 \right) \\ &= \frac{2d(1-\alpha)}{\alpha} \left(\frac{(1-\alpha)^d}{\alpha^d} - 1 \right). \end{aligned}$$

□

Evaluated at $\Delta = 3$, we obtain $\alpha_3 \approx 0.488$. Take $\alpha = 0.5 - c/\Delta^\gamma$, with c a constant not dependent on Δ and $\gamma > 0$. When $\gamma < 2$, we have

$$\lim_{\Delta \to \infty} \frac{2\Delta(0.5 + c/\Delta^\gamma)}{(0.5 - c/\Delta)^\gamma} \left(\frac{(0.5 + c/\Delta^\gamma)^\Delta}{(0.5 - c/\Delta^\gamma)^\Delta} - 1 \right) = \infty.$$

But, when $\gamma = 2$ and $c < \frac{1}{8}$, we have

$$\lim_{\Delta \to \infty} \frac{2\Delta(0.5 + c/\Delta^2)}{(0.5 - c/\Delta)^2} \left(\frac{(0.5 + c/\Delta^2)^\Delta}{(0.5 - c/\Delta^2)^\Delta} - 1 \right) = 8c < 1.$$

Thus, α_Δ is of the form $0.5 - O(1/\Delta^2)$.

3.3 A counting algorithm

In this section, we design an algorithm to approximate $Z(G)$. We first demonstrate how to use the contraction proven in Section 3.2 to design a recursive algorithm which approximates $x(G, v \leftarrow n, \beta)$ with error decaying geometrically in the recursion depth. With this, we will present the approximation algorithm and formalize the argument from Section 2.

3.3.1 Approximating probabilities

In this section, we introduce Algorithm 1, ApproxProb. This algorithm recursively computes an approximation $y(G, v \leftarrow n, \beta)$ of $x(G, v \leftarrow n, \beta)$.

Lemma 17. *Take some arbitrary graph G with degree bounded by Δ, as well as some parameters N, α, β satisfying $\|\nabla f_n^v\|_1 = c < 1$, where f_n^v is the function from equation (4), for each vertex v in every subgraph of G with any β for the particular values of N and α. Pick any computation depth $k > 0$. Algorithm 1 called with parameters G, k, α, N, and β will produce an approximation*

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

Algorithm 1 ApproxProb

1: **Input**
2: $G = (V, E)$ Bounded degree Graph with at least 1 vertex
3: k Remaining Computation Depth
4: v Vertex in the graph G
5: α Value in $[0, 0.5)$
6: n^* Integer value to assign vertex v
7: N Positive Integer
8: β Set of constraints for the vertices V, with β_v denoting the assigned value
9: **Output**
10: $\tilde{f}_n(\mathbf{x})$ Estimate of $f_n(\mathbf{x})$, probability vertex v is assigned n^* in the Gibbs distribution
11: Let $b = \max\{\beta_{u_i}\}$, where u_i is the set of constrained neighbours of v
12: **if** $n^* > N - b$ **then** Return 0
13: **else if** $\exists$ constrained adjacent vertices u_1, u_2 with $\beta_{u_1} + \beta_{u_2} > N$ **then** Return 0
14: **else if** $k = 0$ **then** Return $\frac{1}{\min\{(1-\alpha)N+1, N-b+1\}}$
15: **else**
16: Let d be the number of unconstrained neighbours v has, and let $v_1, \ldots v_d$ be those neighbours
17: **for** $j \in [1, d]$ **do**
18: **for** $n_1, \ldots n_j \in [0, (1-\alpha)N]$ **do**
19: Let $y^j(\bar{n}_j) = \text{ApproxProb}\,(G \setminus v, k-1, v_j, \alpha, n_i, N, (\beta, v_i \leftarrow n_i, \forall i < j))$
20: **end for**
21: **end for**
22: Return $f_n(\mathbf{y})$
23: **end if**

$y = y(G, v \leftarrow n^*, \beta)$ of $x = x(G, v \leftarrow n^*, \beta)$ satisfying

$$|y - x| \leq c^k.$$

Further, it will do so in time $N^{O(k\Delta)}$.

Proof. We will first need to establish that the estimated marginal probabilities $\mathbf{y}$ computed on line 19 form a Feasible Marginal Distribution. We will do so inductively, starting with $k = 1$, as it is the first time we compute such a $\mathbf{y}$. When $k = 1$, $\mathbf{y}$ will form a Feasible Marginal Distribution by construction. The Feasible Probability Distribution requirements will be satisfied by the value returned on line 14, since it will be the uniform probability measure over all feasible values of n^*. Further, for any $j \in \{1, \ldots d\}$ and $\bar{n}_j, \bar{m}_j \in [0, \alpha N]^j$, $y^j(\bar{n}_j) = y^j(\bar{m}_j) = \frac{1}{(1-\alpha)N}$, satisfying the final requirement.

Proceeding by induction for $k > 1$, we have that the Feasible Probability Distribution requirements will be satisfied by Lemma 6 and the inductive hypothesis. Additionally, observe that if

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

$n, m \le \alpha N$, changing $u \leftarrow n$ to $u \leftarrow m$ in a constraint set β will not change the result of running ApproxProb. This is because u assigned to either value cannot violate any constraints. Finally, notice that the vector $\mathbf{y}$ computed in a call to ApproxProb does not depend on n^*, and $f_n = f_m$ for any $n, m \le \alpha N$. Thus, the vector $\mathbf{y}$ computed on line 19 must satisfy equation (13), so $\mathbf{y}$ is a Feasible Marginal Distribution.

We will now show the main result via induction. For the base case, suppose that $k = 1$. Let $\mathbf{x}$ be the true values of the variables $x^i(\bar{n}_i)$ according to the Gibbs distribution of their corresponding graphs. Let $\mathbf{y}$ be the values of the variables computed in the execution of the algorithm on line 19. By the mean value theorem, there must exists some $\mathbf{z}$, which is a linear interpolation of $\mathbf{y}$ and $\mathbf{x}$ such that

$$f_n^v(\mathbf{x}) - f_n^v(\mathbf{y}) = \nabla f_n^v(\mathbf{z}) \cdot (\mathbf{x} - \mathbf{y}).$$

By Lemma 7, $\mathbf{z}$ must also be a Feasible Marginal Distribution. Further, every entry in $\mathbf{x}$ and $\mathbf{y}$ is non-negative and bounded by 1. Thus,

$$|f_n^v(\mathbf{x}) - f_n^v(\mathbf{y})| \le \|\nabla f_n^v(\mathbf{z})\|_1 \|\mathbf{x} - \mathbf{y}\|_\infty \le c.$$

We now induct over the values of k. Once again, let $\mathbf{x}$ be the true values of the variables $x^i(\bar{n}_i)$, and let $\mathbf{y}$ be the values computed on line 19 during the outermost layer of the recursion. Notice that any entries of $\mathbf{y}$ which are 0 will also be 0 in $\mathbf{x}$. This, combined with the inductive hypothesis, implies that $\|\mathbf{x} - \mathbf{y}\|_\infty \le c^{k-1}$. Thus, using the same logic as above, we can use the mean value theorem to show that $|f_n^v(\mathbf{x}) - f_n^v(\mathbf{y})\| \le c^k$ as desired.

Additionally, we wish to analyse the runtime of Algorithm 1. Let $T(k)$ denote the runtime of the algorithm with the input k, and notice that $T(0) = O(1)$. For runs with higher values of k, the limiting time step will be computing the recursive calls on line 20. There will be at most $(1-\alpha)^\Delta N^\Delta$ iterations, and each will take $O(\Delta^2)$ time to update the values of β. Therefore, we get that $T(k) = O((1-\alpha)^\Delta N^\Delta \Delta^2 T(k-1))$ implying that $T(k) = O\left(\left((1-\alpha)^\Delta N^\Delta \Delta^2\right)^k\right)$. Since we are interested in the runtime in terms of Δ, k and N, this gets us that $T(k) = N^{O(k\Delta)}$. □

3.3.2 Counting algorithm

We now introduce Algorithm 2 ApproxZ, which will approximately compute $Z(G)$, the number of restricted relaxed independent sets in the graph G. This algorithm directly computes the telescoping definition of $Z(G)$ using the approximated marginal probabilities produced by Algorithm 1. We will show the following main result.

Theorem 18. *For any graph G of maximum degree Δ with m vertices, Algorithm 2 will produce an estimate $\hat{Z}(G)$ satisfying*

$$1 - O(1/m) \le \frac{\hat{Z}(G)}{Z(G)} \le O(1/m),$$

for any N and any $\alpha > \alpha_\Delta$ for some k which is $O(\log mN)$. Further, the algorithm will run in $N^{O(\Delta \log Nm)}$ time.

Proof. Since $\alpha > \alpha_\Delta$, we have that $\|\nabla f_n\|_1 \le c < 1$ with respect to every vertex v in ever subgraph of G. Observe that equations (10) and (11) together imply that $y(G, v \leftarrow 0, \beta) \ge \frac{1}{(1-\alpha)N+1}$. Thus, letting $k = \log_c\left(\frac{1}{((1-\alpha)\alpha N+1)m^2}\right)$ in the execution of Algorithm 2, Lemma 17 says the estimates $y(G, v_k \leftarrow 0, \beta_{k-1})$ of $x(G, v_k \leftarrow 0, \beta_{k-1})$ will satisfy

$$|x(G, v_k \leftarrow 0, \beta_{k-1}) - y(G, v_k \leftarrow 0, \beta_{k-1})| \le \frac{1}{((1-\alpha)N+1)m^2} \le \frac{y(G, v_k \leftarrow 0, \beta_{k-1})}{m^2}.$$

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

Algorithm 2 ApproxZ

Input
  $G = (V, E)$ Bounded degree Graph with at least 1 vertex
  k Computation Depth
  α Value in $[0, 0.5)$
  N Positive Integer
Output
  $\tilde{Z}(G)$ Estimate of $Z(G)$, the number of restricted relaxed independent sets in G
Let $z_inv = 1$
Let $\beta = ()$
for $v \in V$ **do**
  $z_inv \leftarrow z_inv * \text{ApproxProb}\,(G, k, v, \alpha, 0, N, \beta)$
  $\beta \leftarrow (\beta, v \leftarrow 0)$
end for
Return $1/z_inv$

Dividing both sides of the equation by $y(G, v_k \leftarrow 0, \beta_{k-1})$ results in

$$\left|\frac{x(G, v_k \leftarrow 0, \beta_{k-1})}{y(G, v_k \leftarrow 0, \beta_{k-1})} - 1\right| \leq \frac{1}{m^2}.$$

As discussed in Section 2, this bound is sufficient to show that the estimate $\hat{Z}(G)$ satisfies

$$1 - O(1/m) \leq \frac{\hat{Z}(G)}{Z(G)} \leq O(1/m).$$

We finish by noting that since $c < 1$, k is $O(\log mN)$ for constant α. The bound on the running time follows directly from the bound in Lemma 17. □

3.4 From counting to volumes

Now that we have an algorithm to count restricted relaxed independent sets, we move to showing how one can approximate the volume of $\mathcal{P}_{\text{IS}}(G, \alpha)$ using this discrete approximation.

Fix an arbitrary graph $G = (V, E)$ with $V = \{v_1, \ldots v_m\}$, where each vertex has degree at most Δ. We define a set of graphs $G_n = (\{v_1, \ldots v_n\}, E_n)$ where $E_n = \{(v_i, v_j) \in E | i, j \leq n\}$, so that $G_m = G$. Additionally, fix $\alpha \geq \alpha_\Delta$ and N. Since we will be inducting over the size of the graph, we will redefine β to be a set of constraints of the form $x_i \leq c$, and have β_i denote the tightest constraint on x_i. In particular, meaning any $\mathbf{x}$ satisfying every constraint in β satisfies $x_i \leq \beta_i$. Similarly, we will redefine $v_i \leftarrow c$ to be the set of constraints $\{x_j \leq N - c | (v_i, v_j) \in E\}$. We assume that $\alpha N \leq \beta_i \leq (1 - \alpha)N$. Under this setup, we define the following family of polytopes,

$$\mathbb{Q}(G_n, \alpha, \beta) = \{\mathbf{x} \in \mathbb{R}^n : 0 \leq x_i \leq \beta_i \; \forall i \in [1, n], x_i + x_j \leq N \; \forall (v_i, v_j) \in E_n\}.$$

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

We will show that for any such polytope, we can approximate its volume by counting the number of restricted relaxed independent sets in G. To do this, we define the following two sets,

$$\mathcal{U}(G_n, \alpha, \beta) = \{\mathbf{x} \in \mathbb{Z}^n : 0 \le x_i \le \beta_i \ \forall i \in [1, n], x_i + x_j \le N \ \forall (v_i, v_j) \in E_n\},$$
$$\mathcal{L}(G, \alpha, \beta) = \{\mathbf{x} \in \mathbb{Z}^n : 0 \le x_i \le \beta_i - 1 \ \forall i \in [1, n], x_i + x_j \le N - 2 \ \forall (v_i, v_j) \in E_n\}.$$

Notice that $\frac{\mathbb{Q}(G,\alpha,\beta)}{N^m} \subseteq \mathcal{P}_{\mathrm{IS}}(G, \alpha)$, and when β is such that $\beta_i = (1-\alpha)N, \forall i$, $|\mathcal{U}(G, \alpha, \beta)| = Z(G)$ and $\frac{\mathbb{Q}(G,\alpha,\beta)}{N^m} = \mathcal{P}_{\mathrm{IS}}(G, \alpha)$.

Lemma 19 (Volume bound). *For any graph G and any n, β, α, N,*

$$|\mathcal{L}(G, \alpha, \beta)| \le \mathrm{Vol}\,(\mathbb{Q}(G_n, \alpha, \beta)) \le |\mathcal{U}(G_n, \alpha, \beta)|.$$

Proof. We first show the lower bound. Take any vector $\mathbf{x} \in \mathcal{L}(G, \alpha, \beta)$. It follows immediately from construction that $\mathbf{x} \in \mathbb{Q}(G_n, \alpha, \beta)$. Further, For any vector $\mathbf{y} \in [0, 1)^n$, we have that $x_i + y_i \le \beta_i$ and $(x_i + y_i) + (x_j + y_j) \le x_i + x_j + 2 \le N$ for every $(i, j) \in E_n$. Thus, $\mathbf{x} + \mathbf{y} \in \mathbb{Q}(G_n, \alpha, \beta)$ for every such $\mathbf{y}$. Since we can associate a distinct unit cube within $\mathbb{Q}(G_n, \alpha, \beta)$ to each point $\mathbf{x} \in \mathcal{L}(G, \alpha, \beta)$, we get that $|\mathcal{L}(G, \alpha, \beta)| \le \mathrm{Vol}\,(\mathbb{Q}(G_n, \alpha, \beta))$ as desired.

Next we show the upper bound. Consider any vector $\mathbf{y} \in \mathbb{Q}(G_n, \alpha, \beta)$, and let $\lfloor \mathbf{y} \rfloor$ be the vector $\mathbf{y}$ with each entry truncated to the highest integer value from below. We clearly have $\lfloor \mathbf{y} \rfloor \in \mathcal{U}(G_n, \alpha, \beta)$. Assigning a cube of side length one to each $\mathbf{x} \in \mathcal{U}(G_n, \alpha, \beta)$, we obtain a space which contains $\mathbb{Q}(G_n, \alpha, \beta)$. Thus, $\mathrm{Vol}\,(\mathbb{Q}(G_n, \alpha, \beta)) \le |\mathcal{U}(G_n, \alpha, \beta)|$. □

Now, we move on to the main theorem of this section.

Theorem 20 (Volume approximation). *For any graph G and any of n, β, α, N,*

$$\left(1 - \frac{n}{\alpha N}\right) |\mathcal{U}(G_n, \alpha, \beta)| \le \mathrm{Vol}\,(\mathbb{Q}(G_n, \alpha, \beta)) \le |\mathcal{U}(G_n, \alpha, \beta)|.$$

Proof. Let $\mathbf{x} \in \mathcal{U}(G_n, \alpha, \beta)$ be picked uniformly at random. Further, let $\mu_\beta^n = \mathbb{P}(\mathbf{x} \notin \mathcal{L}(G, \alpha, \beta)) = \frac{|\mathcal{U}(G_n,\alpha,\beta)| - |\mathcal{L}(G,\alpha,\beta)|}{|\mathcal{U}(G_n,\alpha,\beta)|}$. We will show $\mu_\beta^n \le \frac{n}{\alpha N}$ by induction on n. This will directly imply the result by Lemma 19 since $|\mathcal{L}(G_n, \alpha, \beta)| = (1 - \mu_\beta^n)|\mathcal{U}(G_n, \alpha, \beta)|$. In the base case, if $n = 1$, then $\mu_\beta^1 = \mathbb{P}(x_1 \leftarrow \beta_1) = \frac{1}{\beta_1 + 1} \le \frac{1}{\alpha N}$, since x_1 is uniformly distributed over $[0, \beta_1]$ and $\beta_1 \ge \alpha N$.

We now proceed by induction. First, notice that for any $\mathbf{x} \in \mathcal{U}(G_n, \alpha, \beta)$, replacing the nth entry of $\mathbf{x}$ with any smaller value will produce a new $\mathbf{x}' \in \mathcal{U}(G_n, \alpha, \beta)$. In particular, this means for any $0 \le j \le i$, it must be that $\mathbb{P}(x_n \leftarrow i) \le \mathbb{P}(x_n \leftarrow j)$. Thus,

$$\mathbb{P}(x_n \leftarrow i) \le \frac{\mathbb{P}(x_n \leftarrow i)}{\sum_{j=0}^{i} \mathbb{P}(x_n \leftarrow j)} \le \frac{\mathbb{P}(x_n \leftarrow i)}{(i+1)\mathbb{P}(x_n \leftarrow i)} = \frac{1}{i+1},$$

and since $\beta_n \ge \alpha N$, we get that $\mathbb{P}(x_n \leftarrow \beta_i) \le \frac{1}{\alpha N}$.

Next, consider any $\mathbf{x} \in \mathcal{U}(G_n, \alpha, \beta)$ with $x_n = k$ where $\mathbf{x} \notin \mathcal{L}(G, \alpha, \beta)$. It must be that either $x_n = \beta_n$ or $(x_1, \ldots x_{n-1}) \notin \mathcal{L}(G_{n-1}, \alpha, (\beta, v_n \leftarrow k))$. Notice that $k \le (1 - \alpha)N$, since $\beta_n \le (1 - \alpha)N$ by assumption. Therefore, we can use the inductive hypothesis to see

$$\begin{aligned}
\mu_\beta^n &= \mathbb{P}(x_n \leftarrow \beta_n) + \sum_{i=0}^{\beta_i - 1} \mathbb{P}(x_n \leftarrow i)\mu_{(\beta, x_n \leftarrow i)}^{n-1} \\
&\le \frac{1}{\alpha N} + \sum_{i=0}^{\beta_i - 1} \mathbb{P}(x_n \leftarrow i)\frac{n-1}{\alpha N} \\
&\le \frac{n}{\alpha N}.
\end{aligned}$$

□

With this analysis, we are ready to complete the proof of our main result, Theorem 1.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

Proof of Theorem 1. We will actually prove an asymptotically stronger result, namely that there is an algorithm running in $m^{O(\log m)}$ time which produces a value $\tilde{V}$ satisfying

$$1 - O(1/m) \leq \frac{\tilde{V}}{\mathrm{Vol}\,(\mathcal{P}_{\mathrm{IS}}(G,\alpha))} \leq 1 + O(1/m)$$

when G is an m node graph with a maximum degree of Δ and $\alpha > \alpha_\Delta$. The proof which follows can be modified to a proof of the main theorem by setting N to be $O(m/\epsilon)$ and modifying Theorem 18 to have k be $O(\log m/\epsilon)$.

We start by setting $N = \frac{m^2}{\alpha}$. Let β be such that $\beta_i = (1-\alpha)N$ for every i. We have that $\mathcal{P}_{\mathrm{IS}}(G,\alpha) = \frac{\mathbb{Q}(G,\alpha,\beta)}{N^m}$. By Theorem 20, we have

$$\left(1 - \frac{m}{\alpha N}\right)|\mathcal{U}(G,\alpha,\beta)| = \left(1 - \frac{1}{m}\right)|\mathcal{U}(G,\alpha,\beta)| \leq \mathrm{Vol}\,(\mathbb{Q}(G,\alpha,\beta)) \leq |\mathcal{U}(G,\alpha,\beta)|,$$

as a result

$$\begin{aligned}|\mathcal{U}(G,\alpha,\beta)| - \mathrm{Vol}\,(\mathbb{Q}(G,\alpha,\beta))| &= |\mathcal{U}(G,\alpha,\beta)| - \mathrm{Vol}\,(\mathbb{Q}(G,\alpha,\beta)) \\ &\leq \frac{|\mathcal{U}(G,\alpha,\beta)|}{m} \\ &\leq \frac{2\,\mathrm{Vol}\,(\mathbb{Q}(G,\alpha,\beta))}{m},\end{aligned}$$

where the last inequality follows from $1 - \frac{1}{m} \geq \frac{1}{2}$ (unless the graph is trivial) implying that $2\,\mathrm{Vol}\,(\mathbb{Q}(G,\alpha,\beta)) \geq |\mathcal{U}(G,\alpha,\beta)|$. Additionally, we have that $|\mathcal{U}(G,\alpha,\beta)| = Z(G)$ by construction. Theorem 18 implies the existence of a deterministic algorithm to produce an estimate $\tilde{Z}(G)$ satisfying $|\tilde{Z}(G) - |\mathcal{U}(G,\alpha,\beta)|| \leq c|\mathcal{U}(G,\alpha,\beta)|/m$ for some universal constant c. Thus, we get

$$\begin{aligned}|\,\mathrm{Vol}\,(\mathbb{Q}(G,\alpha,\beta)) - \tilde{Z}(G,\alpha)| &\leq |\,\mathrm{Vol}\,(\mathbb{Q}(G,\alpha,\beta)) - |\mathcal{U}(G,\alpha,\beta)|| + ||\mathcal{U}(G,\alpha,\beta)| - \tilde{Z}(G,\alpha)| \\ &\leq \frac{2}{m}\,\mathrm{Vol}\,(\mathbb{Q}(G,\alpha,\beta)) + \frac{c}{m}|\mathcal{U}(G,\alpha,\beta)| \\ &\leq \frac{2}{m}\,\mathrm{Vol}\,(\mathbb{Q}(G,\alpha,\beta)) + \frac{2c}{m}\,\mathrm{Vol}\,(\mathbb{Q}(G,\alpha,\beta)) \\ &= \frac{2c+2}{m}\,\mathrm{Vol}\,(\mathbb{Q}(G,\alpha,\beta)),\end{aligned}$$

where the last inequality again holds because $2\,\mathrm{Vol}\,(\mathbb{Q}(G,\alpha,\beta)) \geq |\mathcal{U}(G,\alpha,\beta)|$. Therefore, we can take $\tilde{V} = \frac{\tilde{Z}(G,\alpha)}{N^m}$ to be our estimate of the volume of $\mathcal{P}_{\mathrm{IS}}$ to see

$$|\,\mathrm{Vol}\,(\mathcal{P}_{\mathrm{IS}}) - \frac{\tilde{Z}(G,\alpha)}{N^m}| = \frac{|\,\mathrm{Vol}\,(\mathbb{Q}(G,\alpha,\beta)) - \tilde{Z}(G,\alpha)|}{N^m} \leq \frac{2c+2}{mN^m}\,\mathrm{Vol}\,(\mathbb{Q}(G,\alpha,\beta)) = \frac{2c+2}{m}\,\mathrm{Vol}\,(\mathcal{P}_{\mathrm{IS}}).$$

Additionally, substituting in the value of N, the algorithm from Theorem 18 will run in $m^{O(\Delta \log m)}$ time. □

3.5 Barriers to the $\alpha = 0$ case

In this section, we discuss some challenges we encounter when designing an algorithm to compute the volume of the unrestricted ($\alpha = 0$) independent set polytope. While the gradient bound given in Section 3.2 is loose and some adjustment to the argument may admit an asymptotic improvement on our values of α_Δ, it clearly will not generalize to $\alpha = 0$. In fact, we have been able to show computationally that the values of $|\nabla f_n|_1$, with f_n defined as in equation (4) exceeds 1 in bounded degree graphs when $\alpha = 0$. We set $N = 64$ and take a 4-regular tree of depth 50, and assigned each

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

leaf vertex 0 with probability 1. We were than able to repeatedly apply equation (4) to produce a probability distribution for the assignment of the vertices at each level of the tree. We know that correlation decay holds on trees [16], so the probability distributions quickly converges to a fixed point distribution. At depth 50, the probability of a vertex being assigned a value n had converged within 5 significant decimal places.

We evaluated $|\nabla f_n(\mathbf{x})|_1$ for each n with this probability distribution $\mathbf{x}$. We found that for $n \leq 13$, $|\nabla f_n(\mathbf{x})|_1 > 1$. When we modified N, we found that roughly the same proportion of values n satisfied $|\nabla f_n(\mathbf{x})|_1 > 1$. Increasing the degree of the tree increased the number of n where $|\nabla f_n(\mathbf{x})|_1 > 1$.

Since we initialized each leaf node with the same value, every vertex at a specified depth in our tree will have the same probability distribution over $[0, N]$. Thus, we can modify equation (4) to be only a function of $x_0, \ldots x_N$, where x_i is the probability a vertex of depth one below our current vertex is assigned i. This allows us to construct the $N+1 \times N+1$ matrix M, where $M_{ij} = \frac{\partial f_i(\mathbf{x})}{\partial x_j}$. In all of the cases above, we found that this matrix M had a largest singular value greater than 1, and thus had an operator norm exceeding 1. This seems to suggest that methods of showing correlation decay which relies on showing a universal single step contraction, such as one similar to ([10]) does not succeed. It is possible that methods based on multi-step contraction or based on other single step but non-spectral type of contraction do succeed. We leave it as a very interesting question for further research.

4. Proof of Theorem 3

This section will be dedicated to building up the proof of Theorem 3. We will once again establish some prelimary results and notation before establishing the key Correlation Decay property. We will then present our algorithm to approximate the partition function, and conclude by relating the partition function to our desired integral.

4.1 Preliminary results

Fix a vertex $v \in V$ and a set of constraints β. Let $v_1, \ldots v_{\eta'}$, $\eta' \leq \eta$, be the unconstrained neighbours of v (we will take v_0 to be the vertex v itself). For any sequence $n_1, \ldots n_{\eta'} \in [0, N-1]$ and any $i \in [1, \eta']$, we will use $\bar{n}_i$ to denote $(n_1, \ldots n_i)$. For each such sequence, we can consider the associated marginal probabilities

$$x^i(\bar{n}_i) \triangleq x(G \setminus v, v_i \leftarrow n_i, (\beta, (v_j \leftarrow n_j, j < i))).$$

We will use $\mathbf{x}$ to denote the vector of marginals $(x^i(\bar{n}_i), 1 \leq i \leq \eta', \bar{n}_i \in [0, N-1]^i)$. Notice that for any fixed i and $\bar{n}_{i-1}$, we have

$$\sum_{n_i=0}^{N-1} x^i(\bar{n}_i) = \frac{\sum_{n_i=0}^{N-1} Z(G \setminus v, (\beta, (v_j \leftarrow n_j, j \leq i)))}{Z(G \setminus v, (\beta, (v_j \leftarrow n_j, j < i)))} = \frac{Z(G \setminus v, (\beta, (v_j \leftarrow n_j, j < i)))}{Z(G \setminus v, (\beta, (v_j \leftarrow n_j, j < i)))} = 1,$$

which inductively implies

$$\sum_{n_1, \ldots n_i \in [0, N-1]} \prod_{j=1}^{i} x^j(\bar{n}_i) = 1. \tag{19}$$

We now define the function

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

$$g_n(\mathbf{x}) = \frac{\sum_{n_1,\dots n_{\eta'} \in [0,N-1]} \prod_{j=1}^{\eta'} x^j(\bar{n}_j) \prod_{e \in E, v \in e} \Phi_e(n, \mathbf{n}_{e\setminus v})}{\sum_{n_0,n_1,\dots n_{\eta'} \in [0,N-1]} \prod_{j=1}^{\eta'} x^j(\bar{n}_j) \prod_{e \in E, v \in e} \Phi_e(\mathbf{n}_e)} \tag{20}$$

for every $n \in [0, N-1]$, where $\mathbf{n}_{e\setminus v} = (n_i, v_i \in e, v_i \neq v)$ (in other words, the value in $\bar{n}$ corresponding to all the vertices in e, excluding the root vertex). This function captures the recursive structure of the problem, as we will see when we use the following proposition in the next section.

Proposition 21. *The following relation holds for every $n \in [0, N-1]$ and every β:*

$$x(G, v \leftarrow n, \beta) = g_n(\mathbf{x}).$$

Proof. Starting from equation (8), we see that

$$\begin{aligned}
Z(G,\beta) &= \sum_{\mathbf{n}\in D_\beta} p(\mathbf{n}) \\
&= \sum_{\mathbf{n}\in D_\beta} \prod_{e\in E} \Phi_e(\mathbf{n}_e) \\
&= \sum_{n_0,n_1,\dots n_{\eta'}\in[0,N-1]} \prod_{e\in E, v\in e} \Phi_e(\mathbf{n}_e) \sum_{\mathbf{n}\setminus\{0,\dots\eta'\}\in D_\beta} \prod_{e\in E, v\in e} \Phi_e(\mathbf{n}_e) \\
&= \sum_{n_0,n_1,\dots n_{\eta'}\in[0,N-1]} \prod_{e\in E, v\in e} \Phi_e(\mathbf{n}_e) Z(G\setminus v, (\beta, (v_i \leftarrow n_i, i \in 1, \dots \eta'))) \\
&= \sum_{n_0,n_1,\dots n_{\eta'}\in[0,N-1]} \frac{\prod_{j=1}^{\eta'} Z(G\setminus v, (\beta, (v_i \leftarrow n_i, i \le j)))}{\prod_{j=1}^{\eta'-1} Z(G\setminus v, (\beta, (v_i \leftarrow n_i, i \le j)))} \prod_{e\in E, v\in e} \Phi_e(\mathbf{n}_e) \\
&= \sum_{n_0,n_1,\dots n_{\eta'}\in[0,N-1]} Z(G\setminus v, (\beta, v_1 \leftarrow n_1)) \prod_{j=2}^{\eta'} x^j(\bar{n}_j) \prod_{e\in E, v\in e} \Phi_e(\mathbf{n}_e).
\end{aligned}$$

By a similar line of reasoning we will also get that

$$Z(G, (\beta, v \leftarrow n)) = \sum_{n_1,\dots n_{\eta'}\in[0,N-1]} Z(G\setminus v, (\beta, v_1 \leftarrow n_1)) \prod_{j=2}^{\eta'} x^j(\bar{n}_j) \prod_{e\in E, v\in e} \Phi_e(n, \mathbf{n}_{e\setminus v}).$$

Combining these, we can see that

$$\begin{aligned}
x(G, v \leftarrow n, \beta) &= \frac{Z(G, (\beta, v \leftarrow n))}{Z(G,\beta)} \\
&= \frac{\sum_{n_1,\dots n_{\eta'}\in[0,N-1]} Z(G\setminus v, (\beta, v_1 \leftarrow n_1)) \prod_{j=2}^{\eta'} x^j(\bar{n}_j) \prod_{e\in E, v\in e} \Phi_e(n, \mathbf{n}_{e\setminus v})}{\sum_{n_0,n_1,\dots n_{\eta'}\in[0,N-1]} Z(G\setminus v, (\beta, v_1 \leftarrow n_1)) \prod_{j=2}^{\eta'} x^j(\bar{n}_j) \prod_{e\in E, v\in e} \Phi_e(\mathbf{n}_e)}
\end{aligned}$$

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

$$= \frac{\sum_{n_1,\ldots n_{\eta'} \in [0,N-1]} \prod_{j=1}^{\eta'} x^j(\bar{n}_j) \prod_{e\in E, v\in e} \Phi_e(n, \mathbf{n}_{e\setminus v})}{\sum_{n_0,n_1,\ldots n_{\eta'} \in [0,N-1]} \prod_{j=1}^{\eta'} x^j(\bar{n}_j) \prod_{e\in E, v\in e} \Phi_e(\mathbf{n}_e)},$$

where the last equality holds by dividing the numerator and denominator by $Z(G \setminus v, \beta)$. This completes the proof. □

4.2 Integration correlation decay

This section will be dedicated to establishing the necessary correlation decay property, to be used later in our final algorithm.

4.2.1 Computation of derivatives

In this next section, we will compute bounds on the values of the derivatives of our function g_n. For the purpose of clarity, we define two expression to represent the numerator and denominator of our function g_n,

$$U_n = \sum_{n_1,\ldots n_{\eta'} \in [0,N-1]} \prod_{j=1}^{\eta'} x^j(\bar{n}_j) \prod_{e\in E, v\in e} \Phi_e(n, \mathbf{n}_{e\setminus v})$$

$$L = \sum_{l=0}^{N-1} U_l = \sum_{l=0}^{N-1} \sum_{n_1,\ldots n_{\eta'} \in [0,N-1]} \prod_{j=1}^{\eta'} x^j(\bar{n}_j) \prod_{e\in E, v\in e} \Phi_e(l, \mathbf{n}_{e\setminus v}).$$

First, notice that for any value of $\mathbf{x}$ we have that

$$L \geq \sum_{l=0}^{N-1} \sum_{n_1,\ldots n_{\eta'} \in [0,N-1]} \prod_{j=1}^{\eta'} x^j(\bar{n}_j) \prod_{e\in E, v\in e} c_e = N \prod_{e\in E, v\in e} c_e, \tag{21}$$

by equations (7) and (19). Now, fix some $k \in [1, \eta']$ and some vector $\bar{n}_k \in [0, N-1]^k$. We have

$$\frac{\partial U_n}{\partial x^j(\bar{n}_k)} = \prod_{j=1}^{k-1} x^j(\bar{n}_j) \sum_{n_{k+1},\ldots n_{\eta'} \in [0,N-1]} \prod_{j=k+1}^{\eta'} x^j(\bar{n}_j) \prod_{e\in E, v\in e} \Phi_e(n, \mathbf{n}_{e\setminus v})$$

$$\frac{\partial L}{\partial x^k(\bar{n}_k)} = \sum_{l=0}^{N-1} \frac{\partial U_l}{\partial x^k(\bar{n}_k)} = \sum_{l=0}^{N-1} \prod_{j=1}^{k-1} x^j(\bar{n}_j) \sum_{n_{k+1},\ldots n_{\eta'} \in [0,N-1]} \prod_{j=k+1}^{\eta'} x^j(\bar{n}_j) \prod_{e\in E, v\in e} \Phi_e(l, \mathbf{n}_{e\setminus v}).$$

Using these definitions, we compute

$$\left| L \frac{\partial U_n}{\partial x^(\bar{n}_k)} - U_n \frac{\partial L}{\partial x^k(\bar{n}_k)} \right| x^k(\bar{n}_k)$$

$$= \left| \left[\sum_{l=0}^{N-1} \sum_{m_1,\ldots m_{\eta'}=0}^{N-1} \prod_{j=1}^{\eta'} x^j(\bar{m}_j) \prod_{e\in E, v\in e} \Phi_e(l, \mathbf{m}_{e\setminus v}) \right] \left[\sum_{n_{k+1},\ldots n_{\eta'}=0}^{N-1} \prod_{j=1}^{\eta'} x^j(\bar{n}_j) \prod_{e\in E, v\in e} \Phi_e(n, \mathbf{n}_{e\setminus v}) \right] \right.$$

$$\left. - \left[\sum_{m_1,\ldots m_{\eta'}=0}^{N-1} \prod_{j=1}^{\eta'} x^j(\bar{m}_j) \prod_{e\in E, v\in e} \Phi_e(n, \mathbf{m}_{e\setminus v}) \right] \left[\sum_{l=0}^{N-1} \sum_{n_{k+1},\ldots n_{\eta'}=0}^{N-1} \prod_{j=1}^{\eta'} x^j(\bar{n}_j) \prod_{e\in E, v\in e} \Phi_e(l, \mathbf{n}_{e\setminus v}) \right] \right|$$

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

$$\leq \sum_{l=0}^{N-1} \sum_{m_1,\ldots m_{\eta'}=0}^{N-1} \sum_{n_{k+1},\ldots n_{\eta'}=0}^{N-1} \prod_{j=1}^{\eta'} x^j(\bar{m}_j)x^j(\bar{n}_j)$$

$$\left| \prod_{e\in E, v\in e} \Phi_e(l, \mathbf{m}_{e\setminus v})\Phi_e(n, \mathbf{n}_{e\setminus v}) - \prod_{e\in E, v\in e} \Phi_e(n, \mathbf{m}_{e\setminus v})\Phi_e(l, \mathbf{n}_{e\setminus v}) \right|$$

$$\leq \sum_{l=0}^{N-1} \sum_{m_1,\ldots m_{\eta'}=0}^{N-1} \sum_{n_{k+1},\ldots n_{\eta'}=0}^{N-1} \prod_{j=1}^{\eta'} x^j(\bar{m}_j)x^j(\bar{n}_j) \left(\prod_{e\in E, v\in E} c_e^2 \right) \left((1-\delta) \left(1 + \frac{\log\left(1+\frac{1}{\eta}\right)}{2\Delta} \right)^{2\Delta} - 1 \right)$$

$$\leq \sum_{l=0}^{N-1} \sum_{m_1,\ldots m_{\eta'}=0}^{N-1} \sum_{n_{k+1},\ldots n_{\eta'}=0}^{N-1} \prod_{j=1}^{\eta'} x^j(\bar{m}_j)x^j(\bar{n}_j) \left(\prod_{e\in E, v\in E} c_e^2 \right) (1-\delta) \left(\left(1 + \frac{\log\left(1+\frac{1}{\eta}\right)}{2\Delta} \right)^{2\Delta} - 1 \right)$$

$$\leq (1-\delta)\left(e^{\log(1+1/\eta)} - 1\right) \sum_{m=0}^{N-1} \sum_{m_1,\ldots m_{\eta'}=0}^{N-1} \sum_{n_{k+1},\ldots n_{\eta'}=0}^{N-1} \prod_{j=1}^{\eta'} x^j(\bar{m}_j)x^j(\bar{n}_j) \prod_{e\in E, v\in e} c_e^2$$

$$= \frac{1-\delta}{\eta} \sum_{m=0}^{N-1} \sum_{m_1,\ldots m_{\eta'}=0}^{N-1} \sum_{n_{k+1},\ldots n_{\eta'}=0}^{N-1} \prod_{j=1}^{\eta'} x^j(\bar{m}_j)x^j(\bar{n}_j) \prod_{e\in E, v\in e} c_e^2$$

$$= \frac{N(1-\delta)}{\eta} \prod_{j=1}^{k} x^j(\bar{n}_j) \prod_{e\in E, v\in E} c_e^2,$$

where the second inequality comes from equation (7) and the last equality comes from equation (19). Combining this with equation (21), we get

$$\left| \frac{\partial g_n(\mathbf{x})}{\partial x^k(\bar{n}_k)} \right| = \left| \frac{L \frac{\partial U_n}{\partial x^k(\bar{n}_k)} - U_n \frac{\partial L}{\partial x^k(\bar{n}_k)}}{L^2} \right| \leq \frac{\frac{N(1-\delta)}{\eta} \prod_{j=1}^{k-1} x^j(\bar{n}_j) \prod_{e\in E, v\in E} c_e^2}{\left(N \prod_{e\in E, v\in e} c_e\right)^2} = \frac{(1-\delta) \prod_{j=1}^{k-1} x^j(\bar{n}_j)}{\eta N}. \tag{22}$$

4.2.2 Bounding the gradient

The remainder of this section will be dedicated to proving the following theorem.

Theorem 22 (Bounded gradients). *For any $n \in [0, N-1]$, we have that*

$$\|\nabla g_n\|_1 \leq 1 - \delta.$$

Proof. Fix some $k \in [1, \eta']$. Using equation (22), we define

$$p_k \triangleq \sum_{\bar{n}_k \in [0,N-1]^k} \left| \frac{\partial g_n(\mathbf{x})}{\partial x^k(\bar{n}_k)} \right|,$$

and use equations (22) and (19) to see

$$p_k \leq \sum_{\bar{n}_k \in [0,N-1]^k} \frac{(1-\delta) \prod_{j=1}^{k-1} x^j(\bar{n}_j)}{\eta N} = \sum_{n_k=0}^{N-1} \frac{1-\delta}{\eta N} \sum_{\bar{n}_{k-1} \in [0,N-1]^{k-1}} \prod_{j=1}^{k-1} x^j(\bar{n}_j) = \frac{1-\delta}{\eta}.$$

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

Algorithm 3 ApproxProb'

Input
  Hypergraph G, Constraints β, Recursion Depth d, Assignment n
Output
  Estimate $y(G, v \leftarrow n, \beta)$ of $x(G, v \leftarrow n, \beta)$
if $d = 0$ **then return** $\frac{1}{N}$
else
  Let $v_1, \ldots v_{\eta'}$ be the unconstrained neighbours of v
  for $j \in [1, \eta']$ **do**
    for $n_1, \ldots n_j \in [0, N-1]$ **do**
      Let $y^j(\bar{n}_j) = \text{ApproxProb'}\,(G \setminus v, (\beta, (v_i \leftarrow n_i, i < j)), d-1, n_j)$
    end for
  end for
  return $g_n(\mathbf{y})$ //where g_n is defined in equation (20)
end if

Thus, since the vertex v has $\eta' < \eta$ neighbours, we get that

$$\|\nabla g_n\|_1 = \sum_{k=1}^{\eta'} p_k \leq \sum_{k=1}^{\eta} \frac{1-\delta}{\eta} = 1 - \delta. \qquad \square$$

4.3 Approximation algorithm

This section will be dedicated to the algorithm used to approximate the value $Z(G)$. The primary piece in this algorithm, present in algorithm 3, will be a recursive algorithm to approximate the value of $x(G, v \leftarrow n, \beta)$.

Lemma 23. *Algorithm 3 will produce an estimate $y(G, v \leftarrow n, \beta)$ satisfying*

$$|y(G, v \leftarrow n, \beta) - x(G, v \leftarrow n, \beta)| \leq (1-\delta)^d.$$

This estimate will be computed in time $O(N^{\eta d})$.

Proof. We will prove the above theorem by induction. When $d = 0$, the difference is trivially bounded by 1, so the theorem will hold. For $d > 1$, we have by induction that the coordinates of $\mathbf{y}$ computed in the highest level run of the algorithm differ from the coordinates of $\mathbf{x}$ by at most $(1-\delta)^{d-1}$. Further, by the mean value theorem, there must be some linear interpolation $\hat{\mathbf{x}}$ of $\mathbf{y}$ and $\mathbf{x}$ which satisfies

$$g_n(\mathbf{y}) - g_n(\mathbf{x}) = \nabla g_n(\hat{\mathbf{x}}) \cdot (\mathbf{y} - \mathbf{x}).$$

Thus, we can use theorem 22 to conclude that

$$|g_n(\mathbf{y}) - g_n(\mathbf{x})| \leq \|\nabla g_n(\hat{\mathbf{x}})\|_1 \|(\mathbf{y} - \mathbf{x})\|_\infty \leq (1-\delta)^d.$$

Additionally, we wish to analyse the runtime of algorithm 3. We will let $T(d)$ denote the runtime of the algorithm with recursion depth d, and observe that $T(0) = O(1)$. For higher values

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

of d, the limiting step will be the runtime of the recursive calls. There will be exactly $N^{\eta'}$ such calls, and each one will take at most $O(\eta')$ time to update the value of β. Thus, we get that $T(d) = N^{\eta'} O(\eta') T(d-1)$, implying that $T(d) = O(N^{\eta d})$, since $\eta' \leq \eta$. □

Before we analyse the full algorithm, we will make an observation about the function g_n. In particular, we will require the following Lemma.

Lemma 24. *For any vector of marginal probabilities* $\mathbf{x}$*, we have*

$$g_n(\mathbf{x}) \geq \frac{1}{\sqrt{2}N},$$

as long as the underlying hypergraph G is connected.

Proof. So long as the input vector $\mathbf{x}$ forms a probability distribution, we use equation (20) to see

$$\begin{aligned}
g_n(\mathbf{x}) &= \frac{\sum_{n_1,\ldots n_{\eta'} \in [0,N-1]} \prod_{j=1}^{\eta'} x^j(\bar{n}_j) \prod_{e \in E, v \in e} \Phi_e(n, \mathbf{n}_{e \setminus v})}{\sum_{n_0,n_1,\ldots n_{\eta'} \in [0,N-1]} \prod_{j=1}^{\eta'} x^j(\bar{n}_j) \prod_{e \in E, v \in e} \Phi_e(\mathbf{n}_e)} \\
&\geq \frac{\sum_{n_1,\ldots n_{\eta'} \in [0,N-1]} \prod_{j=1}^{\eta'} x^j(\bar{n}_j) \prod_{e \in E, v \in e} c_e}{\sum_{n_0,n_1,\ldots n_{\eta'} \in [0,N-1]} \prod_{j=1}^{\eta'} x^j(\bar{n}_j) \prod_{e \in E, v \in e} c_e \left(1 + \frac{\log(1+\frac{1}{\eta})}{2\Delta}\right)} \\
&\geq \frac{\sum_{n_1,\ldots n_{\eta'} \in [0,N-1]} \prod_{j=1}^{\eta'} x^j(\bar{n}_j)}{\left(1 + \frac{\log(1+\frac{1}{\eta})}{2\Delta}\right)^{\Delta} \sum_{n_0,n_1,\ldots n_{\eta'} \in [0,N-1]} \prod_{j=1}^{\eta'} x^j(\bar{n}_j)} \\
&= \frac{1}{N\left(1 + \frac{\log(1+\frac{1}{\eta})}{2\Delta}\right)^{\Delta}},
\end{aligned}$$

where the first inequality follows from the assumption in equation (7), the second inequality follows from v being a member of at most Δ edges, and the final equality follows from equation (19). Further, we can observe that

$$\left(1 + \frac{\log(1+\frac{1}{\eta})}{2\Delta}\right)^{\Delta} \leq \exp\left(\log\left(1+\frac{1}{\eta}\right)/2\right) = \sqrt{1+\frac{1}{\eta}} \leq \sqrt{2},$$

so long as the hypergraph is connected. Thus,

$$g_n(\mathbf{x}) \geq \frac{1}{\sqrt{2}N}.$$ □

With this Lemma, we conclude with the following theorem.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

Algorithm 4 ApproxZ'

Input
  Hypergraph G, Recursion Depth d
Output
  Estimate $\tilde{Z}(G)$ of $Z(G)$
Let $z_inv \leftarrow 1$ and $\beta \leftarrow \{\}$
for $v \in V$ **do**
  $z_inv \leftarrow z_inv * \mathsf{ApproxProb}'(\mathsf{G}, \beta, \mathsf{d}, 0)$
  $\beta \leftarrow (\beta, v \leftarrow 0)$
end for
return $p(\mathbf{0})/z_inv$

Theorem 25. *The estimate $\tilde{Z}(G)$ of $Z(G)$ produced by algorithm 4 will satisfy*

$$1 - O(1/m) \leq \frac{\tilde{Z}(G)}{Z(G)} \leq 1 + O(1/m)$$

for some $d = O(\log(Nm))$. This estimate will be computed in time $N^{O(\eta \log(Nm))}$.

Proof. Order the vertices v_1 to v_m in the same order they are iterated in algorithm 4, and let $\beta_i = (v_j \leftarrow 0, j \leq i)$ as in the end of Section 2. Set $d = \log_{1-\delta}(\frac{1}{\sqrt{2}Nm^2})$. This will result in our approximations $y(G, v_i \leftarrow 0, \beta_{i-1})$ computed in the algorithm to satisfy

$$|x(G, v_i \leftarrow 0, \beta_{i-1}) - y(G, v_i \leftarrow 0, \beta_{i-1})| \leq (1-\delta)^d = \frac{1}{\sqrt{2}Nm^2} \leq \frac{y(G, v_i \leftarrow 0, \beta_{i-1})}{m^2}$$

by Lemma 23 and 24. Dividing both sides by $y(G, v_i \leftarrow 0, \beta_{i-1})$, we get

$$\left| \frac{x(G, v_i \leftarrow 0, \beta_{i-1})}{y(G, v_i \leftarrow 0, \beta_{i-1})} - 1 \right| \leq \frac{1}{m^2}.$$

Next, notice that

$$\frac{\tilde{Z}(G)}{Z(G)} = \frac{p(\mathbf{0}) \prod_{i=1}^{n} y^{-1}(G, v_i \leftarrow 0)}{p(\mathbf{0}) \prod_{i=1}^{n} x^{-1}(G, v_i \leftarrow 0)} = \frac{\prod_{i=1}^{n} x(G, v_i \leftarrow 0)}{\prod_{i=1}^{n} y(G, v_i \leftarrow 0)},$$

by equation (9). Therefore, the estimate $\tilde{Z}$ produced by algorithm 4 will satisfy

$$\left(1 - \frac{1}{m^2}\right)^m = 1 - O(1/m) \leq \frac{\tilde{Z}(G)}{Z(G)} \leq 1 + O(1/m) \leq \left(1 + \frac{1}{m^2}\right)^m.$$

The runtime of this algorithm will be dominated by the calls to the function ApproxProb', which each take time $O(N^{\eta d})$ by Lemma 23. Further, notice that $d = O(\log(Nm))$ since $(1-\delta) < 1$. Thus, since there are m calls to this function with this d, the algorithm will take time $N^{O(\eta \log(Nm))}$. □

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

4.4 From discrete to continuous approximations

In this section, we will present one method for relating the value of a Riemann sum to the value of the corresponding integral, based on bounding the functions $l2$-norm. This process will conclude with a short proof of theorem 3.

Lemma 26. *Suppose we have a function f defined as in equation (5), where each function Φ_e is differentiable and satisfies $\|\nabla\Phi_e(\mathbf{x})\|_2 \le kc_e$ for every $\mathbf{x} \in [0,1]^m$. Then,*

$$\left(1 - \frac{\sqrt{m}k}{N}\right)^{|E|} \le \frac{\int 1(\mathbf{n} \in [0,1]^m)p(\mathbf{n})d\lambda}{Z(G)/N^m} \le \left(1 + \frac{\sqrt{m}k}{N}\right)^{|E|},$$

where $Z(G)$ is defined as in equation (6).

Proof. Pick any vector $\mathbf{n} \in \mathcal{D}$, and let $\mathbf{n}'$ be any vector satisfying $n_i \le n_i' < n_i + 1/N$ for each index i. We will denote the set of all such $\mathbf{n}'$ as $\mathcal{D}_{\mathbf{n}}$. Notice that $\|\mathbf{n} - \mathbf{n}'\|_2 \le \frac{\sqrt{m}}{N}$. Thus, for every edge $e \in E$, we have

$$\Phi_e(\mathbf{n}) - \frac{\sqrt{m}}{N}kc_e \le \Phi_e(\mathbf{n}') \le \Phi_e(\mathbf{n}) + \frac{\sqrt{m}}{N}kc_e.$$

Using this, we see

$$\frac{\int_{\mathcal{D}_{\mathbf{n}}} p(\mathbf{x})d\mathbf{x}}{p(\mathbf{n})/N^m} \le \frac{\int_{\mathcal{D}_{\mathbf{n}}} \prod_{e\in E}\left(\Phi_e(\mathbf{n}) + \frac{\sqrt{m}}{N}kc_e d\mathbf{x}\right)}{\prod_{e\in E}\Phi_e(\mathbf{n})/N^m} = \frac{\prod_{e\in E}\left(\Phi_e(\mathbf{n}) + \frac{\sqrt{m}}{N}kc_e\right)}{\prod_{e\in E}\Phi_e(\mathbf{n})} \le \left(1 + \frac{\sqrt{m}k}{N}\right)^{|E|},$$

where the first inequality follows from the above equation, and the second inequality follows from the assumption in equation (7). Similar reasoning will yield that $\frac{\int_{\mathcal{D}_{\mathbf{n}}} p(\mathbf{x})d\mathbf{x}}{p(\mathbf{n})/N^m} \ge \left(1 - \frac{\sqrt{m}k}{N}\right)^{|E|}$. Finally, we observe that

$$\frac{\int 1(\mathbf{n} \in [0,1]^m)p(\mathbf{n})d\lambda}{Z(G)/N^m} = \frac{\sum_{\mathbf{n}\in\mathcal{D}} \int_{\mathcal{D}_{\mathbf{n}}} p(\mathbf{x})d\mathbf{x}}{\sum_{\mathbf{n}\in\mathcal{D}} p(\mathbf{n})/N^m} \le \max_{\mathbf{n}\in\mathcal{D}} \frac{\int_{\mathcal{D}_{\mathbf{n}}} p(\mathbf{x})d\mathbf{x}}{p(\mathbf{n})/N^m} \le \left(1 + \frac{\sqrt{m}k}{N}\right)^{|E|},$$

with similar logic yielding

$$\frac{\int 1(\mathbf{n} \in [0,1]^m)p(\mathbf{n})d\lambda}{Z(G)/N^m} \ge \left(1 - \frac{\sqrt{m}k}{N}\right)^{|E|}.$$

□

We are now ready to prove our second main result.

Proof of Theorem 3

Let $N = km^{3/2}|E|$. By Lemma 26, we will have that

$$e^{-1/m} \le \frac{\int 1(\mathbf{n} \in [0,1]^m)p(\mathbf{n})d\lambda}{Z(G)/N^m} \le e^{1/m},$$

implying that

$$1 - O(1/m) \le \frac{Z(G)/N^m}{\int 1(\mathbf{n} \in [0,1]^m)p(\mathbf{n})d\lambda} \le 1 + O(1/m).$$

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

Now, let $\tilde{Z}(G)$ be the estimate of $Z(G)$ produced by algorithm 4 when $d = O(\log(Nm)) = O(\log(mk|E|))$. By theorem 25, we must have that

$$1 - O(1/m) \leq \frac{\tilde{Z}(G)}{Z(G)} \leq 1 + O(1/m).$$

Finally, we observe that

$$\left(\frac{Z(G)/N^m}{\int 1(\mathbf{n} \in [0,1]^m)p(\mathbf{n})d\lambda}\right)\left(\frac{\tilde{Z}(G)}{Z(G)}\right) = \frac{\tilde{Z}(G)/N^m}{\int 1(\mathbf{n} \in [0,1]^m)p(\mathbf{n})d\lambda},$$

which means

$$(1 - O(1/m))^2 = 1 - O(1/m) \leq \frac{\tilde{Z}(G)/N^m}{\int 1(\mathbf{n} \in [0,1]^m)p(\mathbf{n})d\lambda} \leq 1 + O(1/m) = (1 + O(1/m))^2.$$

Thus, if we set our estimate of the integral $\tilde{V} = \tilde{Z}(G)/N^m$, we will have that

$$\left|\frac{\tilde{V}}{\int 1(\mathbf{n} \in [0,1]^m)p(\mathbf{n})d\lambda} - 1\right| \leq O(1/m).$$

Further, the runtime to compute this estimate will be dominated by the runtime of algorithm 4. By Theorem 25, this will be $(km|E|)^{O(\eta \log(km|E|))}$. Observing $|E| \leq \Delta m$ and taking η, k, and Δ to be constants, we get a runtime of $m^{O(\log m)}$.

References

[1] Anari, N., Liu, Ku. and Gharan, S. O. (2024) Spectral independence in high-dimensional expanders and applications to the hardcore model. *SIAM J. Comput.* **53**(6) FOCS20-1–FOCS20-37.

[2] Bandyopadhyay, A. and Gamarnik, D. (2006) Counting without sampling. New algorithms for enumeration problems using statistical physics. In Proceedings of 17th ACM-SIAM Symposium on Discrete Algorithms (SODA).

[3] Bandyopadhyay, A. and Gamarnik, D. (2008) Counting without sampling. Asymptotics of the log-partition function for certain statistical physics models. *Random Struct. Algorithms* **33**(4) 452–479.

[4] Bárány, I. and Füredy, Z. (1987) Computing the volume is difficult. *Discrete Comput. Geom.* **2**(4) 319–326.

[5] Barvinok, A. (2016) Combinatorics and complexity of partition functions. Springer.

[6] Barvinok, A. (2024a) Integrating products of quadratic forms. *Discrete Comput Geom.* **72**(2) 603–621.

[7] Barvinok, A. (2024b) On the zeros of partition functions with multi-spin interactions. arXiv preprint arXiv:2406.04179.

[8] Barvinok, A. I. (1994) A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Math. Oper. Res.* **19**(4) 769–779.

[9] Bencs, F. and Regts, G. (2024) Approximating the volume of a truncated relaxation of the independence polytope. arXiv preprint arXiv:2404.08577.

[10] Chen, Z., Liu, K., Mani, N. and Moitra, A. (2023) Strong spatial mixing for colorings on trees and its algorithmic applications. arXiv preprint arXiv:2304.01954.

[11] Duane, S., Kennedy, A. D., Pendleton, B. J. and Roweth, D. (1987) Hybrid Monte Carlo. *Phys. Lett. B* **195**(2) 216–222.

[12] Dyer, M., Frieze, A. and Kannan, R. (1991) A random polynomial time algorithm for approximating the volume of convex bodies. *J. Assoc. Comput. Mach.* **38** 1–17.

[13] Dyer, M. E. and Frieze, A. M. (1988) On the complexity of computing the volume of a polyhedron. *SIAM J. Comput.* **17**(5) 967–974.

[14] Gamarnik, D. and Katz, D. (2012) Correlation decay and deterministic FPTAS for counting list-colorings of a graph. *J. Discrete Algorithms* **12** 29–47.

[15] Gamarnik, D. (2023) Correlation decay and the absence of zeros property of partition functions. *Random Struct. Algorithms* **62** 155–180.

[16] Gamarnik, D. and Ramanan, K. (2019) Uniqueness of Gibbs measures for continuous hardcore models. *Ann. Probab.* **47**(4) 1949–1981.

[17] Gelfand, A. E. and Smith, A. F. M. (1990) Sampling-based approaches to calculating marginal densities. *J. Am. Stat. Assoc.* **85**(410) 398–409.

[18] Guo, H. and Vishvajeet, N. (2025) Deterministic approximation for the volume of the truncated fractional matching polytope. In *16th Innovations in Theoretical Computer Science Conference (ITCS 2025)*, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, pp. 1–57.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558

[19] Hastings, W. K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**(1) 97–109.
[20] Lawrence, J. (1991) Polytope volume computation. *Math. Comput.* **57**(195) 259–271.
[21] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953) Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**(6) 1087–1092.
[22] Mou, W., Flammarion, N., Wainwright, M. J. and Bartlett, P. L. (2022) An efficient sampling algorithm for non-smooth composite potentials. *J. Mach. Learn. Res.* **23**(1) 10611–10660.
[23] Neal, R. (2011) MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo* (Brooks, S., Gelman, A., Jones, G., & Meng, X.-L., eds), CRC Press.
[24] Patel, V. and Regts, G. (2017) Deterministic polynomial-time approximation algorithms for partition functions and graph polynomials. *SIAM J. Comput.* **46**(6) 1893–1919.
[25] Regts, G. (2023) Absence of zeros implies strong spatial mixing. *Probab. Theory Relat. Fields.* **186**(1-2) 621–641.
[26] Roberts, G. O. and Tweedie, R. L. (1996) Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli* **2**(4) 341–363.
[27] Sly, A. (2010) Computational transition at the uniqueness threshold. In *2010 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, IEEE, pp. 287–296.
[28] Trevisan, L. (2006) Pseudorandomness and combinatorial constructions. In Proceedings oh the International Congress of Mathematicians: Madrid, August 22-30, 2006: Invited lectures.
[29] Weitz, D. (2006) Counting independent sets up to the tree threshold. In Proc. 38th Ann. Symposium on the Theory of Computing.

Cite this article: Gamarnik D and Smedira D (2026). Computing volume of a polytope and integrals in high dimensions deterministically. *Combinatorics, Probability and Computing*. https://doi.org/10.1017/S0963548326100558

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 15:10:54, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0963548326100558