A Dichotomy for Bounded Degree Graph Homomorphisms with Nonnegative Weights

Artem Govorov[1]
Department of Computer Sciences, University of Wisconsin-Madison, Madison, WI, USA
hovarau@cs.wisc.edu

Jin-Yi Cai
Department of Computer Sciences, University of Wisconsin-Madison, Madison, WI, USA
jyc@cs.wisc.edu

Martin Dyer
School of Computing, University of Leeds, UK
M.E.Dyer@leeds.ac.uk

Abstract

We consider the complexity of counting weighted graph homomorphisms defined by a symmetric matrix A. Each symmetric matrix A defines a graph homomorphism function $Z_A(\cdot)$, also known as the partition function. Dyer and Greenhill [10] established a complexity dichotomy of $Z_A(\cdot)$ for symmetric $\{0,1\}$-matrices A, and they further proved that its #P-hardness part also holds for bounded degree graphs. Bulatov and Grohe [4] extended the Dyer-Greenhill dichotomy to nonnegative symmetric matrices A. However, their hardness proof requires graphs of arbitrarily large degree, and whether the bounded degree part of the Dyer-Greenhill dichotomy can be extended has been an open problem for 15 years. We resolve this open problem and prove that for nonnegative symmetric A, either $Z_A(G)$ is in polynomial time for all graphs G, or it is #P-hard for bounded degree (and simple) graphs G. We further extend the complexity dichotomy to include nonnegative vertex weights. Additionally, we prove that the #P-hardness part of the dichotomy by Goldberg et al. [12] for $Z_A(\cdot)$ also holds for simple graphs, where A is any real symmetric matrix.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases Graph homomorphism, Complexity dichotomy, Counting problems

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.66

Category Track A: Algorithms, Complexity and Games

Related Version `https://arxiv.org/abs/2002.02021`

Funding *Artem Govorov*: Supported by NSF CCF-1714275.
Jin-Yi Cai: Supported by NSF CCF-1714275.
Martin Dyer: Supported by EPSRC grant EP/S016562/1.

1 Introduction

The modern study of graph homomorphisms originates from the work by Lovász and others several decades ago and has been a very active area [18, 14]. If G and H are two graphs, a graph homomorphism (GH) is a mapping $f\colon V(G) \to V(H)$ that preserves vertex adjacency, i.e., whenever (u, v) is an edge in G, $(f(u), f(v))$ is also an edge in H. Many combinatorial problems on graphs can be expressed as graph homomorphism problems. Well-known examples include the problems of finding a proper vertex coloring, vertex cover, independent

[1] Artem Govorov is the author's preferred spelling of his name, rather than the official spelling Artsiom Hovarau.

47th International Colloquium on Automata, Languages, and Programming (ICALP 2020).
Editors: Artur Czumaj, Anuj Dawar, and Emanuela Merelli; Article No. 66; pp. 66:1–66:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

set and clique. For example, if $V(H) = \{0, 1\}$ with an edge between 0 and 1 and a loop at 0, then $f\colon V(G) \to \{0, 1\}$ is a graph homomorphism iff $f^{-1}(1)$ is an independent set in G; similarly, proper vertex colorings on G using at most m colors correspond to homomorphisms from G to $H = K_m$ (with no loops).

More generally, one can consider weighted graphs H and aggregate all homomorphisms from G to H into a weighted sum. This is a powerful graph invariant which can express many graph properties. Formally, for a symmetric $m \times m$ matrix A, the *graph homomorphism function* on a graph $G = (V, E)$ is defined as follows:

$$Z_A(G) = \sum_{\xi : V \to [m]} \prod_{(u,v) \in E} A_{\xi(u),\xi(v)}.$$

Note that if H is unweighted, and A is its $\{0, 1\}$-adjacency matrix, then each product $\prod_{(u,v)\in E} A_{\xi(u),\xi(v)}$ is 0 or 1, and is 1 iff ξ is a graph homomorphism. Thus in this case $Z_A(G)$ counts the number of homomorphisms from G to H. One can further allow H to have vertex weights. In this case, we can similarly define the function $Z_{A,D}(\cdot)$ (see Definition 4).

These sum-of-product functions $Z_A(\cdot)$ and $Z_{A,D}(\cdot)$ are referred to as the *partition functions* in statistical physics [3]. Various special cases of GH have been studied there extensively, which include the Ising, Potts, hardcore gas, Beach, Widom-Rowlinsom models, etc. [3].

The computational complexity of $Z_A(\cdot)$ has been studied systematically. Dyer and Greenhill [10, 11] proved that, for a symmetric $\{0, 1\}$-matrix A, $Z_A(\cdot)$ is either in polynomial time or #P-complete, and they gave a succinct condition for this complexity dichotomy: if A satisfies the condition then $Z_A(\cdot)$ is computable in polynomial time (we also call it *tractable*), otherwise it is #P-complete. Bulatov and Grohe [4] (see also [22, 13]) generalized the Dyer-Greenhill dichotomy to $Z_A(\cdot)$ for nonnegative symmetric matrices A. It was further extended by Goldberg et al. [12] to arbitrary real symmetric matrices, and finally by Cai, Chen and Lu [7] to arbitrary complex symmetric matrices. In the last two dichotomies, the tractability criteria are not trivial to state. Nevertheless, both tractability criteria are decidable in polynomial time (in the size of A).

The definition of the partition function $Z_A(\cdot)$ can be easily extended to directed graphs G and arbitrary (not necessarily symmetric) matrices A corresponding to directed edge weighted graphs H. Concerning the complexity of counting directed GH, we currently have the *decidable* dichotomies by Dyer, Goldberg and Paterson [9] for $\{0, 1\}$-matrices corresponding to (unweighted) simple acyclic graphs H, and by Cai and Chen [6] for all nonnegative matrices A.

Dyer and Greenhill in the same paper [10] proved a stronger statement that if a $\{0, 1\}$-matrix A fails the tractability condition then $Z_A(G)$ is #P-complete even when restricted to bounded degree graphs G. We note that the complexity of GH for bounded degree graphs is particularly interesting as much work has been done on the approximate complexity of GH focused on bounded degree graphs and approximate algorithms are achieved for them [8, 25, 21, 20, 17, 1, 2, 19, 15]. However, for fifteen years the worst case complexity for bounded degree graphs in the Bulatov-Grohe dichotomy was open. Since this dichotomy is used essentially in almost all subsequent work, e.g., [12, 7], this has been a stumbling block.

Our main contribution in this paper is to resolve this 15-year-old open problem. We prove that the #P-hardness part of the Bulatov-Grohe dichotomy still holds for *bounded degree graphs*. It can be further strengthened to apply to bounded degree *simple* graphs. We actually prove a broader dichotomy for $Z_{A,D}(\cdot)$, where in addition to the nonnegative symmetric edge weight matrix A there is also a nonnegative diagonal vertex weight matrix D. We will give an explicit tractability condition such that, if (A, D) satisfies the condition then $Z_{A,D}(G)$ is computable in polynomial time for all G, and if it fails the condition then $Z_{A,D}(G)$ is #P-hard even restricted to *bounded degree simple graphs* G. $Z_A(G)$ is the special case of $Z_{A,D}(G)$ when D is the identity matrix. Additionally, we prove that the #P-hardness part of the dichotomy by Goldberg et al. [12] for all real symmetric edge weight matrices A still holds for *simple graphs*. (Although in this case, whether under the same condition on A the #P-hardness still holds for bounded degree graphs is not resolved in the present paper.)

In order to prove the dichotomy theorem on bounded degree graphs, we have to introduce a nontrivial extension of the well-developed interpolation method [24]. We use some of the well-established techniques in this area of research such as stretchings and thickenings. But the main innovation is an overall design of the interpolation for a more abstract target polynomial than $Z_{A,D}$. To carry out the proof there is an initial condensation step where we combine vertices that have proportionately the same neighboring edge weights (technically defined by pairwise linear dependence) into a super vertex with a combined vertex weight. Note that this creates vertex weights even when initially all vertex weights are 1. When vertex weights are present, an approach in interpolation proof is to arrange things well so that in the end one can redistribute vertex weights to edge weights. However, when edge weights are not 0-1, any gadget design must deal with a quantity at each vertex that cannot be directly *redistributed*. This dependence has the form $\sum_{j=1}^{m_{\zeta(w)}} \alpha_{\zeta(w)j}\mu_{\zeta(w)j}^{\deg(w)}$, resulting from combining pairwise linearly dependent rows and columns, that depends on the vertex degree $\deg(w)$ in a complicated way. (We note that in the 0-1 case all $\mu_{\zeta(w)j} \in \{0,1\}$, making it in fact degree *independent*.)

We overcome this difficulty by essentially introducing a virtual level of interpolation – an interpolation to realize some "virtual gadget" that cannot be physically realized, and yet its "virtual" vertex weights are suitable for redistribution. Technically we have to define an auxiliary graph G', and express the partition function in an extended framework, called $Z_{\mathscr{A},\mathscr{D}}$ on G' (see Definition 6). In a typical interpolation proof, there is a polynomial with coefficients that have a clear combinatorial meaning defined in terms of G, usually consisting of certain sums of exponentially many terms in some target partition function. Here, we will define a target polynomial with certain coefficients; however these coefficients do not have a direct combinatorial meaning in terms of $Z_{A,D}(G)$, but rather they only have a direct combinatorial meaning in terms of $Z_{\mathscr{A},\mathscr{D}}$ on G'. In a suitable "limiting" sense, a certain aggregate of these coefficients forms some useful quantity in the final result. This introduces a concomitant "virtual" vertex weight which depends on the vertex degree that is "just-right" so that it can be redistributed to become part of the incident edge weight, thus effectively killing the vertex weight. This leads to a reduction from $Z_C(\cdot)$ (without vertex weight) to $Z_{A,D}(\cdot)$, for some C that inherits the hardness condition of A, thus proving the #P-hardness of the latter. This high level description will be made clearer in Section 4. The nature of the degree dependent vertex weight introduces a substantial difficulty; in particular a direct adaptation of the proof in [10] does not work.

Our extended vertex-weighted version of the Bulatov-Grohe dichotomy can be used to correct a crucial gap in the proof by Thurley [23] for a dichotomy for $Z_A(\cdot)$ with Hermitian edge weight matrices A, where this degree dependence was also at the root of the difficulty. [2]

[2] In [23], the proof of Lemma 4.22 uses Lemma 4.24. In Lemma 4.24, A is assumed to have pairwise linearly independent rows while Lemma 4.22 does not assume this, and the author appeals to a twin reduction step in [10]. However, unlike in the 0-1 case [10], such a step incurs degree dependent vertex weights. This gap is fixed by our Theorem 8.

2 Preliminaries

In order to state all our complexity results in the strict notion of Turing computability, we adopt the standard model [16] of computation for partition functions, and require that all numbers be from an arbitrary but fixed algebraic extension of $\mathbb{Q}$. We use $\mathbb{R}$ and $\mathbb{C}$ to denote the sets of real and complex algebraic numbers. Many statements remain true in other fields or rings if arithmetic operations can be carried out efficiently in a model of computation (see [5] for more discussions on this issue).

For a positive integer n, we use $[n]$ to denote the set $\{1, \ldots, n\}$. When $n = 0$, $[0] = \emptyset$. We use $[m : n]$, where $m \leq n$, to denote $\{m, m+1, \ldots, n\}$.

In this paper, we consider undirected graphs unless stated otherwise. Following standard definitions, the graph G is allowed to have multiple edges but no loops. (However, we will touch on this issue a few times when G is allowed to have loops.) The graph H can have multiple edges and loops, or more generally, edge weights. For the graph H, we treat its loops as edges.

An edge-weighted graph H on m vertices can be identified with a symmetric $m \times m$ matrix A in the obvious way. We write this correspondence by $H = H_A$ and $A = A_H$.

▶ **Definition 1.** *Let $A \in \mathbb{C}^{m \times m}$ be a symmetric matrix. The problem* EVAL(A) *is defined as follows: Given an undirected graph $G = (V, E)$, compute*

$$Z_A(G) = \sum_{\xi : V \to [m]} \prod_{(u,v) \in E} A_{\xi(u),\xi(v)}.$$

The function $Z_A(\cdot)$ is called a *graph homomorphism function* or a *partition function*. When A is a symmetric $\{0,1\}$-matrix, i.e., when the graph $H = H_A$ is unweighted, $Z_A(G)$ counts the number of homomorphisms from G to H. In this case, we denote EVAL(H) = EVAL(A_H), and this problem is also known as the $\#H$-coloring problem.

▶ **Theorem 2** (Dyer and Greenhill [10]). *Let H be a fixed undirected graph. Then* EVAL(H) *is in polynomial time if every connected component of H is either (1) an isolated vertex, or (2) a complete graph with all loops present, or (3) a complete bipartite graph with no loops present. Otherwise, the problem* EVAL(H) *is #P-complete.*

Bulatov and Grohe [4] extended Theorem 2 to EVAL(A) where A is a symmetric matrix with nonnegative entries. In order to state their result, we need to define a few notions first.

We say a nonnegative symmetric $m \times m$ matrix A is rectangular if there are pairwise disjoint nonempty subsets of $[m]$: $T_1, \ldots, T_r, P_1, \ldots, P_s, Q_1, \ldots, Q_s$, for some $r, s \geq 0$, such that $A_{i,j} > 0$ iff

$$(i, j) \in \bigcup_{k \in [r]} (T_k \times T_k) \cup \bigcup_{l \in [s]} [(P_l \times Q_l) \cup (Q_l \times P_l)].$$

We refer to $T_k \times T_k, P_l \times Q_l$ and $Q_l \times P_l$ as blocks of A. Further, we say a nonnegative symmetric matrix A is *block-rank*-1 if A is rectangular and every block of A has rank one.

▶ **Theorem 3** (Bulatov and Grohe [4]). *Let A be a symmetric matrix with nonnegative entries. Then* EVAL(A) *is in polynomial time if A is block-rank-1, and is #P-hard otherwise.*

There is a natural extension of EVAL(A) involving the use of vertex weights. Both papers [10, 4] use them in their proofs. A graph H on m vertices with vertex and edge weights is identified with a symmetric $m \times m$ edge weight matrix A and a diagonal $m \times m$ vertex weight matrix $D = \text{diag}(D_1, \ldots, D_m)$ in a natural way. Then the problem EVAL(A) can be generalized to EVAL(A, D) for vertex-edge-weighted graphs.

▶ **Definition 4.** *Let $A \in \mathbb{C}^{m\times m}$ be a symmetric matrix and $D \in \mathbb{C}^{m\times m}$ a diagonal matrix. The problem* $\text{EVAL}(A, D)$ *is defined as follows: Given an undirected graph $G = (V, E)$, compute*

$$Z_{A,D}(G) = \sum_{\xi: V\to[m]} \prod_{w\in V} D_{\xi(w)} \prod_{(u,v)\in E} A_{\xi(u),\xi(v)}.$$

Note that $\text{EVAL}(A)$ is the special case $\text{EVAL}(A, I_m)$. We also need to define another EVAL problem where the vertex weights are specified by the degree.

▶ **Definition 5.** *Let $A \in \mathbb{C}^{m\times m}$ be a symmetric matrix and $\mathfrak{D} = \{D^{[\![i]\!]}\}_{i=0}^{\infty}$ a sequence of diagonal matrices in $\mathbb{C}^{m\times m}$. The problem* $\text{EVAL}(A, \mathfrak{D})$ *is defined as follows: Given an undirected graph $G = (V, E)$, compute*

$$Z_{A,\mathfrak{D}}(G) = \sum_{\xi: V\to[m]} \prod_{w\in V} D^{[\![\deg(w)]\!]}_{\xi(w)} \prod_{(u,v)\in E} A_{\xi(u),\xi(v)}.$$

Finally, we need to define a general EVAL problem, where the vertices and edges can individually take specific weights. Let $\mathscr{A}$ be a set of (edge weight) $m \times m$ matrices and $\mathscr{D}$ a set of diagonal (vertex weight) $m \times m$ matrices. A GH-grid $\Omega = (G, \rho)$ consists of a graph $G = (V, E)$ with possibly both directed and undirected edges, and loops, and ρ assigns to each edge $e \in E$ or loop an $A^{(e)} \in \mathscr{A}$ and to each vertex $v \in V$ a $D^{(v)} \in \mathscr{D}$. (A loop is just an edge of the form (v, v).) If $e \in E$ is a directed edge then the tail and head correspond to rows and columns of $A^{(e)}$, respectively; if $e \in E$ is an undirected edge then $A^{(e)}$ must be symmetric.

▶ **Definition 6.** *The problem* $\text{EVAL}(\mathscr{A}, \mathscr{D})$ *is defined as follows: Given a GH-grid $\Omega = \Omega(G)$, compute*

$$Z_{\mathscr{A},\mathscr{D}}(\Omega) = \sum_{\xi:\, V\to[m]} \prod_{w\in V} D^{(w)}_{\xi(w)} \prod_{e=(u,v)\in E} A^{(e)}_{\xi(u),\xi(v)}$$

We remark that $Z_{\mathscr{A},\mathscr{D}}$ is introduced only as a tool to express a certain quantity in a "virtual" interpolation; the dichotomy theorems do not apply to this. Definitions 5 and 6 are carefully crafted in order to carry out the #P-hardness part of the proof of Theorem 8. Notice that the problem $\text{EVAL}(\mathscr{A}, \mathscr{D})$ generalizes both problems $\text{EVAL}(A)$ and $\text{EVAL}(A, D)$, by taking $\mathscr{A}$ to be a single symmetric matrix, and by taking $\mathscr{D}$ to be a single diagonal matrix. But $\text{EVAL}(A, \mathfrak{D})$ is not naturally expressible as $\text{EVAL}(\mathscr{A}, \mathscr{D})$ because the latter does not force the vertex-weight matrix on a vertex according to its degree.

We refer to $[m]$ as the domain of the corresponding EVAL problem. If $\mathscr{A} = \{A\}$ or $\mathscr{D} = \{D\}$, then we simply write $Z_{A,\mathscr{D}}(\cdot)$ or $Z_{\mathscr{A},D}(\cdot)$, respectively.

We use a superscript (Δ) and/or a subscript simp to denote the restriction of a corresponding EVAL problem to degree-Δ bounded graphs and/or simple graphs. E.g., $\text{EVAL}^{(\Delta)}(A)$ denotes the problem $\text{EVAL}(A)$ restricted to degree-Δ bounded graphs, $\text{EVAL}_{\text{simp}}(A, \mathfrak{D})$ denotes the problem $\text{EVAL}(A, \mathfrak{D})$ restricted to simple graphs, and both restrictions apply in $\text{EVAL}^{(\Delta)}_{\text{simp}}(A, \mathfrak{D})$.

Working within the framework of $\text{EVAL}(A, D)$, we define an edge gadget to be a graph with two distinguished vertices, called u^* and v^*. An edge gadget $G = (V, E)$ has a signature (edge weight matrix) expressed by an $m \times m$ matrix F, where

$$F_{ij} = \sum_{\substack{\xi:\, V\to[m] \\ \xi(u^*)=i,\, \xi(v^*)=j}} \prod_{z\in V\setminus\{u^*,v^*\}} D_{\xi(z)} \prod_{(x,y)\in E} A_{\xi(x),\xi(y)}$$

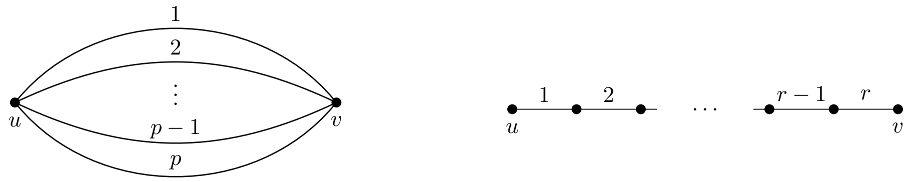


Figure 1 The thickening T_pe and the stretching S_re of an edge $e = (u, v)$.

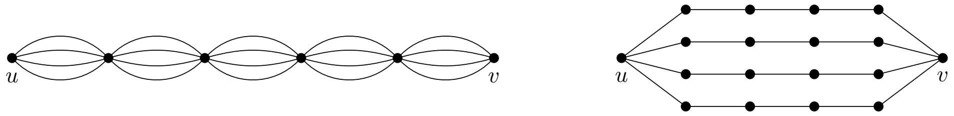


Figure 2 The graphs T_4S_5e (on the left) and S_5T_4e (on the right) where $e = (u, v)$.

for $1 \le i, j \le m$. When this gadget is placed in a graph identifying u^* and v^* with two vertices u and v in that graph, then F is the signature matrix for the pair (u, v). Note that the vertex weights corresponding to u and v are excluded from the product in the definition of F. Similar definitions can be introduced for $\mathrm{EVAL}(A)$, $\mathrm{EVAL}(A, \mathfrak{D})$ and $\mathrm{EVAL}(\mathscr{A}, \mathscr{D})$.

We use $\le_{\mathrm{T}}^{\mathrm{P}}$ (and $\equiv_{\mathrm{T}}^{\mathrm{P}}$) to denote polynomial-time Turing reductions (and equivalences, respectively).

Two simple operations are known as *thickening* and *stretching*. Let $p, r \ge 1$ be integers. A *p-thickening* of an edge replaces it by p parallel edges, and a *r-stretching* replaces it by a path of length r. In both cases we retain the endpoints u, v. The *p-thickening* or *r-stretching* of G with respect to $F \subseteq E(G)$, denoted respectively by $T_p^{(F)}(G)$ and $S_r^{(F)}(G)$, are obtained by *p-thickening* or *r-stretching* each edge from F, respectively. Other edges, if any, are unchanged in both cases. When $F = E(G)$, we call them the *p-thickening* and *r-stretching* of G and denote them by $T_p(G)$ and $S_r(G)$, respectively. T_pe and S_re are the special cases when the graph consists of a single edge e. See Figure 1 for an illustration. Thickenings and stretchings can be combined in any order. Examples are shown in Figure 2.

For a matrix A, we denote by $A^{\odot p}$ the matrix obtained by replacing each entry of A with its pth power. Clearly, $Z_A(T_pG) = Z_{A^{\odot p}}(G)$ and $Z_A(S_rG) = Z_{A^r}(G)$. More generally, for the vertex-weighted case, we have $Z_{A,D}(T_pG) = Z_{A^{\odot p},D}(G)$ and $Z_{A,D}(S_rG) = Z_{A(DA)^{r-1},D}(G)$. Here $(DA)^0 = I_m$ if A and D are $m \times m$.

3 Dichotomy for bounded degree graphs

In addition to the Dyer-Greenhill dichotomy (Theorem 2), in the same paper [10] they also proved that the #P-hardness part of their dichotomy holds for bounded degree graphs. The bounded degree case of the Bulatov-Grohe dichotomy (Theorem 3) was left open, and all known proofs [4, 22, 13] of its #P-hardness part require unbounded degree graphs. All subsequent dichotomies that use the Bulatov-Grohe dichotomy, e.g., [12, 7] also explicitly or implicitly (because of their dependence on the Bulatov-Grohe dichotomy) require unbounded degree graphs. In this paper, we extend the #P-hardness part of the Bulatov-Grohe dichotomy to bounded degree graphs.

▶ **Theorem 7.** *Let A be a symmetric nonnegative matrix. If A is not block-rank-1, then for some $\Delta > 0$, the problem $\mathrm{EVAL}^{(\Delta)}(A)$ is #P-hard.*

The degree bound Δ proved in Theorem 7 depends on A, as is the case in Theorem 2. The authors of [10] conjectured that a universal bound $\Delta = 3$ works for Theorem 2; whether a universal bound exists for both Theorems 2 and 7 is open. For general symmetric real or complex A, it is open whether bounded degree versions of the dichotomies in [12] and [7] hold. Xia [26] proved that a universal bound does not exist for complex symmetric matrices A, assuming #P does not collapse to P.

We prove a broader dichotomy than Theorem 7, which also includes arbitrary nonnegative vertex weights.

▶ **Theorem 8.** *Let A and D be $m \times m$ nonnegative matrices, where A is symmetric, and D is diagonal. Let A' be the matrix obtained from A by striking out rows and columns that correspond to 0 entries of D on the diagonal. If A' is block-rank-1, then the problem $\mathrm{EVAL}(A, D)$ is in polynomial time. Otherwise, for some $\Delta > 0$, the problem $\mathrm{EVAL}^{(\Delta)}_{\mathrm{simp}}(A, D)$ is #P-hard.*

Every 0 entry of D on the diagonal effectively nullifies the corresponding domain element in $[m]$, so the problem becomes an equivalent problem on the reduced domain. Thus, for a nonnegative diagonal D, without loss of generality, we may assume the domain has already been reduced so that D is positive diagonal. In what follows, we will make this assumption.

In Section 5, we will prove the tractability part of Theorem 8. This follows easily from known results. In Section 6, we will present two technical lemmas, Lemma 9 and Lemma 10 to be used in Section 4. Finally, in Section 7 we prove Theorem 11, showing that the #P-hardness part of the dichotomy for counting GH by Goldberg et al. [12] for real symmetric matrix (with mixed signs) is also valid for simple graphs.

4 Hardness proof

We proceed to prove the #P-hardness part of Theorem 8. Let A and D be $m \times m$ matrices, where A is nonnegative symmetric but not block-rank-1, and D is positive diagonal. The first step is to eliminate pairwise linearly dependent rows and columns of A. (We will see that this step will naturally create nontrivial vertex weights even if we initially start with the vertex unweighted case $D = I_m$.)

If A has a zero row or column i, then for any connected input graph G other than a single isolated vertex, no map $\xi : V(G) \to [m]$ having a nonzero contribution to $Z_{A,D}(G)$ can map any vertex of G to i. So, by crossing out all zero rows and columns (they have the same index set since A is symmetric) we may assume that A has no zero rows or columns. We then delete the same set of rows and columns from D, thereby expressing the problem $\mathrm{EVAL}^{(\Delta)}_{\mathrm{simp}}(A, D)$ for $\Delta \ge 0$ on a smaller domain. Also permuting the rows and columns of both A and D simultaneously by the same permutation does not change the value of $Z_{A,D}(\cdot)$, and so it does not change the complexity of $\mathrm{EVAL}^{(\Delta)}_{\mathrm{simp}}(A, D)$ for $\Delta \ge 0$ either. Having no zero rows and columns implies that pairwise linear dependence is an equivalence relation, and so we may assume that the pairwise linearly dependent rows and columns of A are contiguously arranged. Then, after renaming the indices, the entries of A are of the following form: $A_{(i,j),(i',j')} = \mu_{ij}\mu_{i'j'}A'_{i,i'}$, where A' is a nonnegative symmetric $s \times s$ matrix with all columns nonzero and pairwise linearly independent, $1 \le i, i' \le s$, $1 \le j \le m_i$, $1 \le j' \le m_{i'}$, $\sum_{i=1}^{s} m_i = m$, and all $\mu_{ij} > 0$. We also rename the indices of the matrix D so that the diagonal entries of D are of the following form: $D_{(i,j),(i,j)} = \alpha_{ij} > 0$ for $1 \le i \le s$ and $1 \le j \le m_i$. As $m \ge 1$ we get $s \ge 1$.

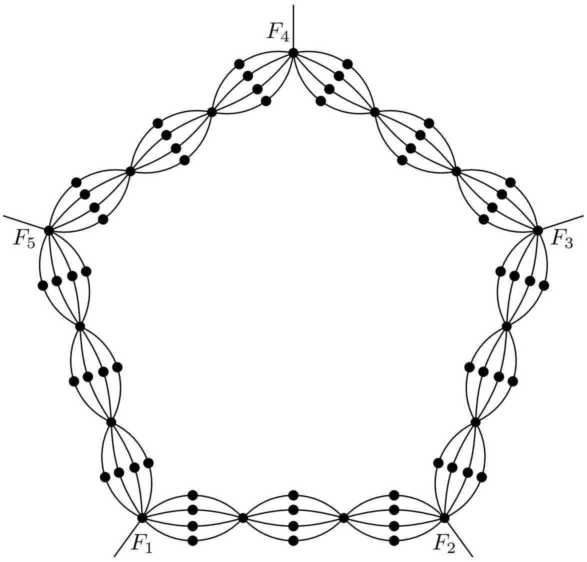


Figure 3 The gadget $\mathcal{R}_{5,3,4}$.

Then the partition function $Z_{A,D}(\cdot)$ can be written in a compressed form

$$Z_{A,D}(G) = \sum_{\zeta:V(G)\to[s]} \left(\prod_{w\in V(G)} \sum_{j=1}^{m_{\zeta(w)}} \alpha_{\zeta(w)j}\mu_{\zeta(w)j}^{\deg(w)} \right) \prod_{(u,v)\in E(G)} A'_{\zeta(u),\zeta(v)} = Z_{A',\mathfrak{D}}(G)$$

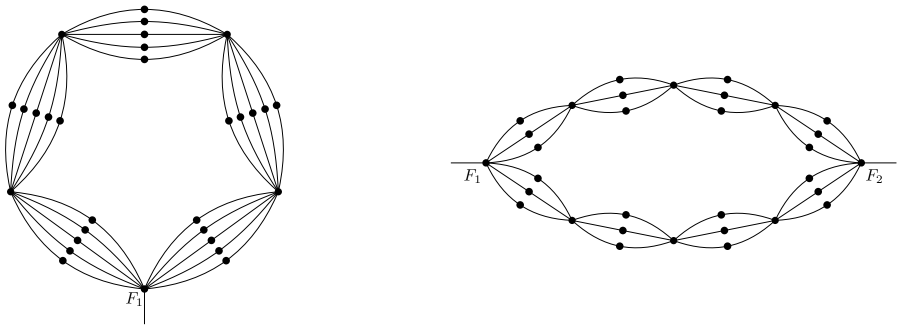
where $\mathfrak{D} = \{D^{[\![k]\!]}\}_{k=0}^{\infty}$ with $D_i^{[\![k]\!]} = \sum_{j=1}^{m_i} \alpha_{ij}\mu_{ij}^k > 0$ for $k \ge 0$ and $1 \le i \le s$. Then all matrices in $\mathfrak{D}$ are positive diagonal. Note the dependence on the vertex degree $\deg(w)$ for $w \in V(G)$. Since the underlying graph G remains unchanged, this way we obtain the equivalence $\text{EVAL}_{\text{simp}}^{(\Delta)}(A, D) \equiv_{\text{T}}^{\text{P}} \text{EVAL}_{\text{simp}}^{(\Delta)}(A', \mathfrak{D})$ for any $\Delta \ge 0$. Here the subscript simp can be included or excluded, and the same is true for the superscript (Δ), the statement remains true in all cases. We also point out that the entries of the matrices $D^{[\![k]\!]} \in \mathfrak{D}$ are computable in polynomial time in the input size of (A, D) as well as in k.

4.1 Gadgets $\mathcal{P}_{n,p}$ and $\mathcal{R}_{d,n,p}$

We first introduce the *edge gadget* $\mathcal{P}_{n,p}$, for all $p, n \ge 1$. It is obtained by replacing each edge of a path of length n by the gadget in Figure 5 from Lemma 10. More succinctly $\mathcal{P}_{n,p}$ is $S_2T_pS_ne$, where e is an edge.

To define the gadget $\mathcal{R}_{d,n,p}$, for all $d, p, n \ge 1$, we start with a cycle on d vertices $F_1, \ldots, F_d$ (call it a d-cycle), replace every edge of the d-cycle by a copy of $\mathcal{P}_{n,p}$, and append a dangling edge at each vertex F_i of the d-cycle. To be specific, a 2-cycle has two vertices with 2 parallel edges between them, and a 1-cycle is a loop on one vertex. The gadget $\mathcal{R}_{d,n,p}$ always has d dangling edges. Note that all $\mathcal{R}_{d,n,p}$ are loopless simple graphs (i.e., without parallel edges or loops), for $d, n, p \ge 1$. An example of a gadget $\mathcal{R}_{d,n,p}$ is shown in Figure 3. For the special cases $d = 1, 2$, examples of gadgets $\mathcal{R}_{d,n,p}$ can be seen in Figure 4.

We note that vertices in $\mathcal{P}_{n,p}$ have degrees at most $2p$, and vertices in $\mathcal{R}_{d,n,p}$ have degrees at most $2p + 1$, taking into account the dangling edges. Clearly $|V(\mathcal{R}_{d,n,p})| = dn(p + 1)$ and $|E(\mathcal{R}_{d,n,p})| = (2np + 1)d$, including the dangling edges.



(a) $\mathcal{R}_{1,5,5}$ **(b)** $\mathcal{R}_{2,4,3}$

Figure 4 Examples of gadgets $\mathcal{R}_{d,n,p}$ for $d = 1, 2$.

By Lemma 10, we can fix some $p \geq 1$ such that $B = (A' D^{\llbracket 2 \rrbracket} A')^{\odot p}$ is nondegenerate, where the superscript $\llbracket 2 \rrbracket$ is from the stretching operator S_2 which creates those degree 2 vertices, and the superscript $\odot p$ is from the thickening operator T_p, followed by S_2, which creates those parallel paths of length 2. The edge gadget $\mathcal{P}_{n,p}$ has the edge weight matrix

$$L^{(n)} = \underbrace{B D^{\llbracket 2p \rrbracket} B \ldots B D^{\llbracket 2p \rrbracket} B}_{D^{\llbracket 2p \rrbracket} \text{ appears } n-1 \,\geq\, 0 \text{ times}} = B (D^{\llbracket 2p \rrbracket} B)^{n-1} \tag{1}$$

$$= (D^{\llbracket 2p \rrbracket})^{-1/2} ((D^{\llbracket 2p \rrbracket})^{1/2} B (D^{\llbracket 2p \rrbracket})^{1/2})^n (D^{\llbracket 2p \rrbracket})^{-1/2}, \tag{2}$$

where in the notation $L^{(n)}$ we suppress the index p. The $n-1$ occurrences of $D^{\llbracket 2p \rrbracket}$ in (1) are due to those $n-1$ vertices of degree $2p$. Here $(D^{\llbracket 2p \rrbracket})^{1/2}$ is a diagonal matrix with the positive square roots of the corresponding entries of $D^{\llbracket 2p \rrbracket}$ on the main diagonal, and $(D^{\llbracket 2p \rrbracket})^{-1/2}$ is its inverse. The vertices F_i are of degree $2p+1$ each, but the contributions by its vertex weights are not included in $L^{(n)}$.

The constraint function induced by $\mathcal{R}_{d,n,p}$ is more complicated to write down. When it is placed as a part of a graph, for any given assignment to the d vertices F_i, we can express the contribution of the gadget $\mathcal{R}_{d,n,p}$ in terms of d copies of $L^{(n)}$, *together with* the vertex weights incurred at the d vertices F_i which will depend on their degrees.

4.2 Interpolation using $\mathcal{R}_{d,n,p}$

Assume for now that G does not contain isolated vertices. We will replace every vertex $u \in V(G)$ of degree $d = d_u = \deg(u) \geq 1$ by a copy of $\mathcal{R}_{d,n,p}$, for all $n, p \geq 1$. The replacement operation can be described in two steps: In step one, each $u \in V(G)$ is replaced by a d-cycle on vertices $F_1, \ldots, F_d$, each having a dangling edge attached. The d dangling edges will be identified one-to-one with the d incident edges at u. If u and v are adjacent vertices in G, then the edge (u, v) in G will be replaced by merging a pair of dangling edges, one from the d_u-cycle and one from the d_v-cycle. Thus in step one we obtain a graph G', which basically replaces every vertex $u \in V(G)$ by a cycle of $\deg(u)$ vertices. Then in step two, for every cycle in G' that corresponds to some $u \in V(G)$ we replace each edge on the cycle by a copy of the edge gadget $\mathcal{P}_{n,p}$.

Let $G_{n,p}$ denote the graph obtained from G by the replacement procedure above. Since all gadgets $\mathcal{R}_{d,n,p}$ are loopless simple graphs, so are $G_{n,p}$ for all $n, p \geq 1$, even if G has multiple edges (or had multiloops, if we view a loop as adding degree 2 to the incident vertex). As a technical remark, if G contains vertices of degree 1, then the intermediate graph G' has loops but all graphs $G_{n,p}$ ($n, p \geq 1$) do not. Also note that all vertices in $G_{n,p}$ have degree at most $2p+1$, which is independent of n.

Next, it is not hard to see that

$$|V(G_{n,p})| = \sum_{u \in V(G)} d_u n(p+1) = 2n(p+1)|E(G)|,$$
$$|E(G_{n,p})| = |E(G)| + \sum_{u \in V(G)} 2npd_u = (4np+1)|E(G)|.$$

Hence the size of the graphs $G_{n,p}$ is polynomially bounded in the size of G, n and p.

Since we chose a fixed p, and will choose n to be bounded by a polynomial in the size of G, whenever something is computable in polynomial time in n, it is also computable in polynomial time in the size of G (we will simply say in polynomial time).

We consider $Z_{A',\mathfrak{D}}(G)$, and substitute G by $G_{n,p}$. We will make use of the edge weight matrix $L^{(n)}$ of $\mathcal{P}_{n,p}$ in (2). The vertices F_i are of degree $2p+1$ each in $G_{n,p}$, so will each contribute a vertex weight according to the diagonal matrix $D^{[\![2p+1]\!]}$ to the partition function, which are not included in $L^{(n)}$, but now must be accounted for in $Z_{A',\mathfrak{D}}(G_{n,p})$.

Since B is real symmetric and $D^{[\![2p]\!]}$ is positive diagonal, the matrix

$$\widetilde{B} = (D^{[\![2p]\!]})^{1/2} B (D^{[\![2p]\!]})^{1/2}$$

is real symmetric. Then $\widetilde{B}$ is orthogonally diagonalizable over $\mathbb{R}$, i.e., there exist a real orthogonal matrix S and a real diagonal matrix $J = \operatorname{diag}(\lambda_i)_{i=1}^{s}$ such that $\widetilde{B} = S^T J S$. Then $\widetilde{B}^n = S^T J^n S$ so the edge weight matrix for $\mathcal{P}_{n,p}$ becomes

$$L^{(n)} = (D^{[\![2p]\!]})^{-1/2} \widetilde{B}^n (D^{[\![2p]\!]})^{-1/2} = (D^{[\![2p]\!]})^{-1/2} S^T J^n S (D^{[\![2p]\!]})^{-1/2}.$$

Note that $L^{(n)}$ as a matrix is defined for any $n \geq 0$, and $L^{(0)} = (D^{[\![2p]\!]})^{-1}$, even though there is no physical gadget $\mathcal{P}_{0,p}$ that corresponds to it. However, it is precisely this "virtual" gadget we wish to "realize" by interpolation.

Clearly, $\widetilde{B}$ is nondegenerate as B and $(D^{[\![2p]\!]})^{1/2}$ both are, and so is J. Then all $\lambda_i \neq 0$. We can also write $L_{ij}^{(n)} = \sum_{\ell=1}^{s} a_{ij\ell} \lambda_\ell^n$ for every $n \geq 0$ and some real $a_{ij\ell}$'s which depend on S, $D^{[\![2p]\!]}$, but not on J and n, for all $1 \leq i, j, \ell \leq s$. By the formal expansion of the symmetric matrix $L^{(n)}$ above, we have $a_{ij\ell} = a_{ji\ell}$. Note that for all $n, p \geq 1$, the gadget $\mathcal{R}_{d_v,n,p}$ for $v \in V(G)$ employs exactly d_v copies of $\mathcal{P}_{n,p}$. Let $t = \sum_{v \in V(G)} d_v = 2|E(G)|$; this is precisely the number of edge gadgets $\mathcal{P}_{n,p}$ in $G_{n,p}$.

In the evaluation of the partition function $Z_{A',\mathfrak{D}}(G_{n,p})$, we stratify the vertex assignments in $G_{n,p}$ as follows. Denote by $\kappa = (k_{ij})_{1 \leq i \leq j \leq s}$ a tuple of nonnegative integers, where the indexing is over all $s(s+1)/2$ ordered pairs (i,j). There are a total of $\binom{t+s(s+1)/2-1}{s(s+1)/2-1}$ such tuples that satisfy $\sum_{1 \leq i \leq j \leq s} k_{ij} = t$. For a fixed s, this is a polynomial in t, and thus a polynomial in the size of G. Denote by $\mathcal{K}$ the set of all such tuples κ. We will stratify all vertex assignments in $G_{n,p}$ by $\kappa \in \mathcal{K}$, namely all assignments such that there are exactly k_{ij} many constituent edge gadgets $\mathcal{P}_{n,p}$ with the two end points (in either order of the end points) assigned i and j respectively.

For each $\kappa \in \mathcal{K}$, the edge gadgets $\mathcal{P}_{n,p}$ in total contribute $\prod_{1 \leq i \leq j \leq s} (L_{ij}^{(n)})^{k_{ij}}$ to the partition function $Z_{A',\mathfrak{D}}(G_{n,p})$. If we factor this product out for each $\kappa \in \mathcal{K}$, we can express $Z_{A',\mathfrak{D}}(G_{n,p})$ as a linear combination of these products over all $\kappa \in \mathcal{K}$, with polynomially many coefficient values c_κ that are independent of all edge gadgets $\mathcal{P}_{n,p}$. Another way to define these coefficients c_κ is to think in terms of G': For any $\kappa = (k_{ij})_{1 \leq i \leq j \leq s} \in \mathcal{K}$, we say a vertex assignment on G' is consistent with κ if it assigns exactly k_{ij} many cycle edges of G' (i.e., those that belong to the cycles that replaced vertices in G) as ordered pairs of vertices to the values (i,j) or (j,i). (For any loop in G', as a cycle of length 1 that came from a degree 1 vertex of G, it can only be assigned (i, i) for some $1 \le i \le s$.) Let L' be any symmetric edge signature to be assigned on each of these cycle edges in G', and keep the edge signature A' on the merged dangling edges between any two such cycles, and the suitable vertex weights specified by $\mathfrak{D}$, namely each vertex receives its vertex weight according to $D^{[\![2p+1]\!]}$. Then c_κ is the sum, over all assignments consistent with κ, of the products of all edge weights and vertex weights *other than* the contributions by L', in the evaluation of the partition function on G'. In other words, for each $\kappa \in \mathcal{K}$,

$$c_\kappa = \sum_{\substack{\zeta\colon V(G') \to [s] \\ \zeta \text{ is consistent with } \kappa}} \prod_{w \in V(G')} D^{[\![2p+1]\!]}_{\zeta(w)} \prod_{(u,v) \in \widetilde{E}} A'_{\zeta(u),\zeta(v)},$$

where $\widetilde{E} \subseteq E(G')$ are the non-cycle edges of G' that are in 1-1 correspondence with $E(G)$.

In particular, the values c_κ are independent of n. Thus for some polynomially many values c_κ, where $\kappa \in \mathcal{K}$, we have for all $n \ge 1$,

$$Z_{A',\mathfrak{D}}(G_{n,p}) = \sum_{\kappa \in \mathcal{K}} c_\kappa \prod_{1 \le i \le j \le s} (L^{(n)}_{ij})^{k_{ij}} = \sum_{\kappa \in \mathcal{K}} c_\kappa \prod_{1 \le i \le j \le s} (\sum_{\ell=1}^{s} a_{ij\ell} \lambda_\ell^n)^{k_{ij}}. \qquad (3)$$

Expanding out the last sum and rearranging the terms, for some values $b_{i_1,\ldots,i_s}$ independent of n, we get

$$Z_{A',\mathfrak{D}}(G_{n,p}) = \sum_{\substack{i_1+\ldots+i_s=t \\ i_1,\ldots,i_s \ge 0}} b_{i_1,\ldots,i_s} (\prod_{j=1}^{s} \lambda_j^{i_j})^n$$

for all $n \ge 1$.

This represents a linear system with the unknowns $b_{i_1,\ldots,i_s}$ with the rows indexed by n. The number of unknowns is clearly $\binom{t+s-1}{s-1}$ which is polynomial in the size of the input graph G since s is a constant. The values $\prod_{j=1}^{s} \lambda_j^{i_j}$ can be clearly computed in polynomial time.

We show how to compute the value

$$\sum_{\substack{i_1+\ldots+i_s=t \\ i_1,\ldots,i_s \ge 0}} b_{i_1,\ldots,i_s}$$

from the values $Z_{A',\mathfrak{D}}(G_{n,p})$, $n \ge 1$ in polynomial time. The coefficient matrix of this system is a Vandermonde matrix. However, it can have repeating columns so it might not be of full rank because the coefficients $\prod_{j=1}^{s} \lambda_j^{i_j}$ do not have to be pairwise distinct. However, when they are equal, say, $\prod_{j=1}^{s} \lambda_j^{i_j} = \prod_{j=1}^{s} \lambda_j^{i'_j}$, we replace the corresponding unknowns $b_{i_1,\ldots,i_s}$ and $b_{i'_1,\ldots,i'_s}$ with their sum as a new variable. Since all $\lambda_i \ne 0$, we have a Vandermonde system of full rank after all such combinations. Therefore we can solve this linear system in polynomial time and find the desired value $\sum_{\substack{i_1+\ldots+i_s=t \\ i_1,\ldots,i_s \ge 0}} b_{i_1,\ldots,i_s}$.

Now we will consider a problem in the framework of $Z_{\mathscr{A},\mathscr{D}}$ according to Definition 6. Let $G_{0,p}$ be the (undirected) GH-grid, with the underlying graph G', and every edge of the cycle in G' corresponding to a vertex in $V(G)$ is assigned the edge weight matrix $(D^{[\![2p]\!]})^{-1}$, and we keep the vertex-weight matrices $D^{[\![2p+1]\!]}$ at all vertices F_i. The other edges, i.e., the original edges of G, each keep the assignment of the edge weight matrix A'. (So in the specification of $Z_{\mathscr{A},\mathscr{D}}$, we have $\mathscr{A} = \{(D^{[\![2p]\!]})^{-1}, A'\}$, and $\mathscr{D} = \{D^{[\![2p+1]\!]}\}$. We note that G' may have loops, and Definition 6 specifically allows this.) Then

$$Z_{\{(D^{[\![2p]\!]})^{-1},A'\},D^{[\![2p+1]\!]}}(G_{0,p}) = \sum_{\substack{i_1+\ldots+i_s=t \\ i_1,\ldots,i_s \ge 0}} b_{i_1,\ldots,i_s} (\prod_{j=1}^{s} \lambda_j^{i_j})^0 = \sum_{\substack{i_1+\ldots+i_s=t \\ i_1,\ldots,i_s \ge 0}} b_{i_1,\ldots,i_s}$$

and we have just computed this value in polynomial time in the size of G from the values $Z_{A',\mathfrak{D}}(G_{n,p})$, for $n \geq 1$. In other words, we have achieved it by querying the oracle $\operatorname{EVAL}(A', \mathfrak{D})$ on the instances $G_{n,p}$, for $n \geq 1$, in polynomial time.

Equivalently, we have shown that we can simulate a virtual "gadget" $\mathcal{R}_{d,0,p}$ replacing every occurrence of $\mathcal{R}_{d,n,p}$ in $G_{n,p}$ in polynomial time. The virtual gadget $\mathcal{R}_{d,0,p}$ has the edge signature $(D^{\llbracket 2p \rrbracket})^{-1}$ in place of $(D^{\llbracket 2p \rrbracket})^{-1/2} \widetilde{B}^n (D^{\llbracket 2p \rrbracket})^{-1/2}$ in each $\mathcal{P}_{n,p}$, since

$$(D^{\llbracket 2p \rrbracket})^{-1/2} \widetilde{B}^0 (D^{\llbracket 2p \rrbracket})^{-1/2} = (D^{\llbracket 2p \rrbracket})^{-1/2} I_s (D^{\llbracket 2p \rrbracket})^{-1/2} = (D^{\llbracket 2p \rrbracket})^{-1}.$$

Additionally, each F_i retains the vertex-weight contribution with the matrix $D^{\llbracket 2p+1 \rrbracket}$ in $\mathcal{R}_{d,0,p}$. We view it as having "virtual" degree $2p+1$. This precisely results in the GH-grid $G_{0,p}$.

However, even though $G_{0,p}$ still retains the cycles, since $(D^{\llbracket 2p \rrbracket})^{-1}$ is a diagonal matrix, each vertex F_i in a cycle is forced to receive the same vertex assignment value in the domain set $[s]$; all other vertex assignments contribute zero in the evaluation of $Z_{\{(D^{\llbracket 2p \rrbracket})^{-1}, A'\}, D^{\llbracket 2p+1 \rrbracket}}(G_{0,p})$. This can be easily seen by traversing the vertices $F_1, \ldots, F_d$ in a cycle. Hence we can view each cycle employing the virtual gadget $\mathcal{R}_{d,0,p}$ as a single vertex that contributes only a diagonal matrix of positive vertex weights $P^{\llbracket d \rrbracket} = (D^{\llbracket 2p+1 \rrbracket} (D^{\llbracket 2p \rrbracket})^{-1})^d$, where d is the vertex degree in G. Contracting all the cycles to a single vertex each, we arrive at the original graph G. Let $\mathfrak{P} = \{P^{\llbracket i \rrbracket}\}_{i=0}^{\infty}$, where we let $P^{\llbracket 0 \rrbracket} = I_s$, and for $i > 0$, we have $P_j^{\llbracket i \rrbracket} = w_j^i$ where $w_j = \sum_{k=1}^{m_j} \alpha_{jk} \mu_{jk}^{2p+1} / \sum_{k=1}^{m_j} \alpha_{jk} \mu_{jk}^{2p} > 0$ for $1 \leq j \leq s$. This shows that we now can interpolate the value $Z_{A',\mathfrak{P}}(G)$ using the values $Z_{A',\mathfrak{D}}(G_{n,p})$ in polynomial time in the size of G. The graph G is arbitrary but without isolated vertices here. We show next how to deal with the case when G has isolated vertices.

Given an arbitrary graph G, assume it has $h \geq 0$ isolated vertices. Let G^* denote the graph obtained from G by their removal. Then G^* is of size not larger than G and $h \leq |V(G)|$. Obviously, $Z_{A',\mathfrak{P}}(G) = (\sum_{i=1}^{s} P_i^{\llbracket 0 \rrbracket})^h Z_{A',\mathfrak{P}}(G^*) = s^h Z_{A',\mathfrak{P}}(G^*)$. Here the integer s is a constant, so the factor $s^h > 0$ can be easily computed in polynomial time. Thus, knowing the value $Z_{A',\mathfrak{P}}(G^*)$ we can compute the value $Z_{A',\mathfrak{P}}(G)$ in polynomial time. Further, since we only use the graphs $G_{n,p}, n \geq 1$ during the interpolation, each being simple of degree at most $2p+1$, combining it with the possible isolated vertex removal step, we conclude $\operatorname{EVAL}(A', \mathfrak{P}) \leq_{\mathrm{T}}^{\mathrm{P}} \operatorname{EVAL}_{\mathrm{simp}}^{(2p+1)}(A', \mathfrak{D})$.

Next, it is easy to see that for an arbitrary graph G

$$\begin{aligned}
Z_{A',\mathfrak{P}}(G) &= \sum_{\zeta: V(G) \to [s]} \prod_{z \in V(G)} P_{\zeta(z)}^{\llbracket \deg(z) \rrbracket} \prod_{(u,v) \in E(G)} A'_{\zeta(u),\zeta(v)} \\
&= \sum_{\zeta: V(G) \to [s]} \prod_{z \in V(G)} w_{\zeta(z)}^{\deg(z)} \prod_{(u,v) \in E(G)} A'_{\zeta(u),\zeta(v)} \\
&= \sum_{\zeta: V(G) \to [s]} \prod_{(u,v) \in E(G)} w_{\zeta(u)} w_{\zeta(v)} A'_{\zeta(u),\zeta(v)} \\
&= \sum_{\zeta: V(G) \to [s]} \prod_{(u,v) \in E(G)} C_{\zeta(u),\zeta(v)} = Z_C(G).
\end{aligned}$$

Here C is an $s \times s$ matrix with the entries $C_{ij} = A'_{ij} w_i w_j$ where $1 \leq i, j \leq s$. Clearly, C is a nonnegative symmetric matrix. In the above chain of equalities, we were able to *redistribute the weights w_i and w_j into the edge weights A'_{ij}* which resulted in the edge weights C_{ij}, so that precisely each edge (u,v) in G gets two factors $w_{\zeta(u)}$ and $w_{\zeta(v)}$ since the vertex weights at u and v were $w_{\zeta(u)}^{\deg(u)}$ and $w_{\zeta(v)}^{\deg(v)}$ respectively. (This is a crucial step in our proof.)

Because the underlying graph G is arbitrary, it follows that $\mathrm{EVAL}(A', \mathfrak{P}) \equiv_{\mathrm{T}}^{\mathrm{P}} \mathrm{EVAL}(C)$. Combining this with the previous EVAL-reductions and equivalences, we obtain

$$\mathrm{EVAL}(C) \equiv_{\mathrm{T}}^{\mathrm{P}} \mathrm{EVAL}(A', \mathfrak{P}) \leq_{\mathrm{T}}^{\mathrm{P}} \mathrm{EVAL}_{\mathrm{simp}}^{(2p+1)}(A', \mathfrak{D}) \equiv_{\mathrm{T}}^{\mathrm{P}} \mathrm{EVAL}_{\mathrm{simp}}^{(2p+1)}(A, D),$$

so that $\mathrm{EVAL}(C) \leq_{\mathrm{T}}^{\mathrm{P}} \mathrm{EVAL}_{\mathrm{simp}}^{(\Delta)}(A, D)$, by taking $\Delta = 2p + 1$.

Remembering that our goal is to prove the #P-hardness for the matrices A, D not satisfying the tractability conditions of Theorem 8, we finally use the assumption that A is not block-rank-1. Next, noticing that all $\mu_{ij} > 0$, by construction A' is not block-rank-1 either. Finally, because all $w_i > 0$ nor is C block-rank-1 implying that $\mathrm{EVAL}(C)$ is #P-hard by Theorem 3. Hence $\mathrm{EVAL}_{\mathrm{simp}}^{(2p+1)}(A, D)$ is also #P-hard. This completes the proof of the #P-hardness part of Theorem 8.

We remark that one important step in our interpolation proof happened at the stratification step before (3). In the proof we have the goal of redistributing vertex weights to edge weights; but this redistribution is sensitive to the degree of the vertices. This led us to define the auxiliary graph G' and the coefficients c_κ. Usually in an interpolation proof there are some coefficients that have a clear combinatorial meaning in terms of the original problem instance. Here these values c_κ do not have a clear combinatorial meaning in terms of $Z_{A',\mathfrak{D}}(G)$, rather they are defined in terms of an intermediate problem instance G', which is neither G nor the actual constructed graphs $G_{n,p}$. It is only in a "limiting" sense that a certain combination of these values c_κ allows us to compute $Z_{A',\mathfrak{D}}(G)$.

5 Tractability part

The tractability part of Theorem 8 follows easily from known results. For completeness we outline a proof here. Let A and D be $m \times m$ matrices, where A is nonnegative symmetric block-rank-1 and D is positive diagonal.

First, $Z_{A,D}(G)$ can be reduced to the connected components $G_1, \ldots, G_t$ of G,

$$Z_{A,D}(G) = \prod_{i=1}^{t} Z_{A,D}(G_i),$$

so we may as well assume G is connected. We permute the rows and columns of A, D by the same permutation and then cross out zero rows and columns of A. This does not change $Z_{A,D}$. We may assume that $A = \mathrm{diag}(A_i)_{i=1}^{k}$ is block diagonal with nonzero blocks $A_1, \ldots, A_k$, where each block A_i is either a symmetric matrix of rank 1 with no zero entries, or a symmetric bipartite matrix of the form $\left(\begin{smallmatrix} 0 & B \\ B^T & 0 \end{smallmatrix}\right)$ where B has rank 1 and no zero entries. Then we can write $D = \mathrm{diag}(D_i)_{i=1}^{k}$ where each D_i is positive diagonal of the corresponding size. As A is block diagonal and G is connected,

$$Z_{A,D}(G) = \sum_{i=1}^{k} Z_{A_i,D_i}(G).$$

So we may as well assume that A is one of these blocks. Also let $D = \mathrm{diag}(\alpha_i)_{i=1}^{m}$.

1) A is a symmetric matrix of rank 1 with no zero entries. We can write $A = x^T x$ for some positive row vector $x = (x_i)_{i=1}^{m}$. Then

$$Z_{A,D}(G) = \prod_{u \in V(G)} \sum_{i=1}^{m} \alpha_i x_i^{\deg(u)}.$$

2) $A = \left(\begin{smallmatrix} 0 & B \\ B^T & 0 \end{smallmatrix}\right)$, where B is $\ell \times (m-\ell)$ (for some $1 \le \ell < m$) has rank 1 and no zero entries. We can write $B = x^T y$ for some positive row vectors $x = (x_i)_{i=1}^{\ell}$ and $y = (y_j)_{j=\ell+1}^{m}$. Since G is connected, $Z_{A,D}(G) = 0$ unless G is bipartite. If G is bipartite with a vertex bipartization $V_1 \cup V_2$, then we only need to consider maps $\xi\colon G \to [m]$ such that either $\xi(V_1) \subseteq [\ell]$, $\xi(V_2) \subseteq [\ell+1:m]$ or $\xi(V_1) \subseteq [\ell+1:m]$, $\xi(V_2) \subseteq [\ell]$, with all other maps contribute zero to $Z_{A,D}(G)$. Then

$$\begin{aligned} Z_{A,D}(G) = {} & \left(\prod_{u\in V_1} \sum_{i=1}^{\ell} \alpha_i x_i^{\deg(u)}\right)\left(\prod_{v\in V_2} \sum_{j=\ell+1}^{m} \alpha_j y_j^{\deg(v)}\right) \\ & + \left(\prod_{u\in V_1} \sum_{j=\ell+1}^{m} \alpha_j y_j^{\deg(u)}\right)\left(\prod_{v\in V_2} \sum_{i=1}^{\ell} \alpha_i x_i^{\deg(v)}\right). \end{aligned}$$

6 Two technical lemmas

We need two technical lemmas. The following lemma is from [10] (Lemma 3.6); for the convenience of readers we give a proof here.

▶ **Lemma 9.** *Let A and D be $m \times m$ matrices, where A is real symmetric with all columns nonzero and pairwise linearly independent, and D is positive diagonal. Then all columns of ADA are nonzero and pairwise linearly independent.*

Proof. The case $m = 1$ is trivial. Assume $m \ge 2$. Let $D = \operatorname{diag}(\alpha_i)_{i=1}^m$, and $\Pi = \operatorname{diag}(\sqrt{\alpha_i})_{i=1}^m$. Then $\Pi^2 = D$. We have $ADA = Q^T Q$, where $Q = \Pi A$. Let q_i denote the ith column of Q. Then Q has pairwise linearly independent columns. By the Cauchy-Schwartz inequality,

$$q_i^T q_j < \left((q_i^T q_i)(q_j^T q_j)\right)^{1/2},$$

whenever $i \ne j$. Then for any $1 \le i < j \le m$, the ith and jth columns of ADA contain a submatrix

$$\begin{bmatrix} q_i^T q_i & q_i^T q_j \\ q_i^T q_j & q_j^T q_j \end{bmatrix},$$

so they are linearly independent. ◀

The following is also adapted from [10] (Theorem 3.1).

▶ **Lemma 10.** *Let A and D be $m \times m$ matrices, where A is real symmetric with all columns nonzero and pairwise linearly independent, and D is positive diagonal. Then for all sufficiently large positive integers p, the matrix $B = (ADA)^{\odot p}$ corresponding to the edge gadget in Figure 5 is nondegenerate.*

Proof. If $m = 1$, then any $p \ge 1$ works. Let $m \ge 2$. Following the proof of Lemma 9, we have $q_i^T q_j < \sqrt{(q_i^T q_i)(q_j^T q_j)}$, for all $1 \le i < j \le m$. Let

$$\gamma = \max_{1\le i<j\le m} \frac{q_i^T q_j}{\sqrt{(q_i^T q_i)(q_j^T q_j)}} < 1.$$

Let $A' = ADA = Q^T Q$ so $A'_{ij} = q_i^T q_j$. Then $A'_{ij} \le \gamma \sqrt{A'_{ii} A'_{jj}}$ for all $i \ne j$. Consider the determinant of A'. Each term of $\det(A')$ has the form

$$\pm \prod_{i=1}^{m} A'_{i\sigma(i)},$$

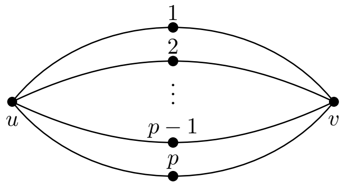


Figure 5 The edge gadget S_2T_pe, $e=(u,v)$ with the edge weight matrix $(ADA)^{\odot p}$.

where σ is a permutation of $[m]$. Denote $t(\sigma)=|\{i \mid \sigma(i)\neq i\}|$. Then

$$\prod_{i=1}^{m} A'_{i\sigma(i)} \le \gamma^{t(\sigma)} \prod_{i=1}^{m}\sqrt{A'_{ii}}\prod_{i=1}^{m}\sqrt{A'_{\sigma(i)\sigma(i)}} = \gamma^{t(\sigma)}\prod_{i=1}^{m} A'_{ii}.$$

Consider the p-thickening of A' for $p\ge 1$. Each term of $\det\left((A')^{\odot p}\right)$ has the form $\pm\prod_{i=1}^{m} {A'}^{\,p}_{i\sigma(i)}$ for some permutation σ of $[m]$. Now

$$|\{\sigma \mid t(\sigma)=j\}| \le \binom{m}{j} j! \le m^j,$$

for $0\le j\le m$. By separating out the identity permutation and all other terms, for $p\ge \lfloor \ln(2m)/\ln(1/\gamma)\rfloor + 1$, we have $2m\gamma^p<1$, and

$$\begin{aligned}\det\left((A')^{\odot p}\right) &\ge \left(\prod_{i=1}^{m} A'_{ii}\right)^p - \left(\prod_{i=1}^{m} A'_{ii}\right)^p \sum_{j=1}^{m} m^j\gamma^{pj}\\ &\ge \left(\prod_{i=1}^{m} A'_{ii}\right)^p\left(1-\frac{m\gamma^p}{1-m\gamma^p}\right) = \left(\prod_{i=1}^{m} A'_{ii}\right)^p\left(\frac{1-2m\gamma^p}{1-m\gamma^p}\right) > 0. \qquad ◀\end{aligned}$$

7 Hardness for $Z_A(\cdot)$ on simple graphs for real symmetric A

There is a more direct approach to prove the #P-hardness part of the Bulatov-Grohe dichotomy (Theorem 3) for simple graphs. Although this method does not handle degree-boundedness, we can apply it more generally to the problem EVAL(A,D) when the matrix A is real symmetric and D is positive diagonal. In particular, we will prove the #P-hardness part of the dichotomy for counting GH by Goldberg et al. [12] (the problem EVAL(A) without vertex weights, where A is a real symmetric matrix) for simple graphs.

We first prove the following theorem.

▶ **Theorem 11.** *Let A and D be $m\times m$ matrices, where A is real symmetric and D is positive diagonal. Then* $\mathrm{EVAL}(A,D) \le_{\mathrm{T}}^{\mathrm{P}} \mathrm{EVAL}_{\mathrm{simp}}(A,D)$.

Proof. We may assume A is not identically 0, for otherwise the problem is trivial. Let $G=(V,E)$ be an input graph to the problem EVAL(A,D). For any $n\ge 1$, let $G_n = S_n^{(F)}(G)$ where $F\subseteq E$ is the subset consisting of the edges of G each of which is parallel to at least one other edge. In other words, we obtain G_n by replacing every parallel edge e by its n-stretching S_ne. We will refer to these as paths of length n in G_n. Note that $G_1=G$. Moreover, for every $n\ge 2$, the graph G_n is simple and loopless, and has polynomial size in the size of G and n.

A path of length $n \geq 1$ has the edge weight matrix

$$M^{(n)} = \underbrace{ADA \dots ADA}_{D \text{ appears } n-1 \,\geq\, 0 \text{ times}} = A(DA)^{n-1} = D^{-1/2}(D^{1/2}AD^{1/2})^n D^{-1/2}.$$

Here $D^{1/2}$ is a diagonal matrix with the positive square roots of the corresponding entries of D on the main diagonal, and $D^{-1/2}$ is its inverse.

Since A is real symmetric and D is positive diagonal, the matrix $\widetilde{A} = D^{1/2}AD^{1/2}$ is real symmetric. Then $\widetilde{A}$ is orthogonally diagonalizable over $\mathbb{R}$, i.e., there exist a real orthogonal matrix S and a real diagonal matrix $J = (\lambda_i)_{i=1}^m$ such that $\widetilde{A} = S^T J S$. If A has rank r, then $1 \leq r \leq m$, and we may assume that $\lambda_i \neq 0$ for $1 \leq i \leq r$ and $\lambda_i = 0$ for $i > r$.

We have $\widetilde{A}^n = S^T J^n S$, so the edge weight matrix for a path of length $n \geq 1$ can be written as

$$M^{(n)} = D^{-1/2}\widetilde{A}^n D^{-1/2} = D^{-1/2} S^T J^n S D^{-1/2}.$$

We can write $M_{ij}^{(n)} = \sum_{\ell=1}^r a_{ij\ell}\lambda_\ell^n$ by a formal expansion, for every $n \geq 1$ and some real $a_{ij\ell}$'s that are dependent on D and S, but independent of n and λ_ℓ, where $1 \leq i, j \leq m$ and $1 \leq \ell \leq r$. By the formal expansion of the symmetric matrix $M^{(n)}$ above, we have $a_{ij\ell} = a_{ji\ell}$. Let $t = |F|$, which is the number of edges in G subject to the stretching operator S_n to form G_n.

In the evaluation of the partition function $Z_{A,D}(G_n)$, we stratify the vertex assignments in G_n as follows. Denote by $\kappa = (k_{ij})_{1 \leq i \leq j \leq m}$ a nonnegative tuple with entries indexed by ordered pairs of nonnegative numbers that satisfy $\sum_{1 \leq i \leq j \leq m} k_{ij} = t$. Let $\mathcal{K}$ denote the set of all such possible tuples κ. In particular, $|\mathcal{K}| = \binom{t+m(m+1)/2-1}{m(m+1)/2-1}$. For a fixed m, this is a polynomial in t, and thus a polynomial in the size of G. Let c_κ be the sum over all assignments of all vertex and edge weight products in $Z_{A,D}(G_n)$, except the contributions by the paths of length n formed by stretching parallel edges in G, such that the endpoints of precisely k_{ij} constituent paths of length n receive the assignments (i, j) (in either order of the end points) for every $1 \leq i \leq j \leq m$. Technically we can call a vertex assignment on G consistent with κ (where $\kappa \in \mathcal{K}$), if it satisfies the stated property. Note that the contribution by each such path does not include the vertex weights of the two end points (but does include all vertex weights of the internal $n-1$ vertices of the path). We can write

$$c_\kappa = \sum_{\substack{\xi\colon V(G) \to [m] \\ \xi \text{ is consistent with } \kappa}} \prod_{w \in V} D_{\xi(w)} \prod_{(u,v) \in E \setminus F} A_{\xi(u),\xi(v)}$$

for $\kappa \in \mathcal{K}$.

In particular, the values c_κ are independent of n. Thus for some polynomially many values c_κ, where $\kappa \in \mathcal{K}$, we have for all $n \geq 1$,

$$Z_{A,D}(G_n) = \sum_{\kappa \in \mathcal{K}} c_\kappa \prod_{1 \leq i \leq j \leq m} (M_{ij}^{(n)})^{k_{ij}} = \sum_{\kappa \in \mathcal{K}} c_\kappa \prod_{1 \leq i \leq j \leq m} (\sum_{\ell=1}^r a_{ij\ell}\lambda_\ell^n)^{k_{ij}}.$$

Expanding out the last sum and rearranging the terms, for some values $b_{i_1,\dots,i_r}$ independent of n, we get

$$Z_{A,D}(G_n) = \sum_{\substack{i_1+\dots+i_r = t \\ i_1,\dots,i_r \geq 0}} b_{i_1,\dots,i_r} (\prod_{\ell=1}^r \lambda_\ell^{i_\ell})^n \tag{4}$$

for all $n \geq 1$.

This can be viewed as a linear system with the unknowns $b_{i_1,\dots,i_r}$ with the rows indexed by n. The number of unknowns is $\binom{t+r-1}{r-1}$ which is polynomial in the size of the input graph G, since $r \le m$ is a constant. The values $\prod_{\ell=1}^{r} \lambda_\ell^{i_\ell}$ can all be computed in polynomial time.

We show how to compute the value $Z_{A,D}(G) = \sum_{\substack{i_1+\ldots+i_r=t\\ i_1,\dots,i_r\ge 0}} b_{i_1,\dots,i_r} \prod_{\ell=1}^{r} \lambda_\ell^{i_\ell}$, from the values $Z_{A,D}(G_n)$ where $n \ge 2$ in polynomial time (recall that G_n is simple and loopless for $n \ge 2$). The coefficient matrix of the linear system (4) is a Vandermonde matrix. However, it might not be of full rank because the coefficients $\prod_{\ell=1}^{r} \lambda_\ell^{i_\ell}$ do not have to be pairwise distinct, and therefore it can have repeating columns. Nevertheless, when there are two repeating columns we replace the corresponding unknowns $b_{i_1,\dots,i_r}$ and $b_{i'_1,\dots,i'_r}$ with their sum as a new variable; we repeat this replacement procedure until there are no repeating columns. Since all $\lambda_\ell \ne 0$, for $1 \le \ell \le r$, after the replacement, we have a Vandermonde system of full rank. Therefore we can solve this modified linear system in polynomial time. This allows us to obtain the value $Z_{A,D}(G) = \sum_{\substack{i_1+\ldots+i_r=t\\ i_1,\dots,i_r\ge 0}} b_{i_1,\dots,i_r} \prod_{\ell=1}^{r} \lambda_\ell^{i_\ell}$, which also has exactly the same pattern of repeating multipliers $\prod_{\ell=1}^{r} \lambda_\ell^{i_\ell}$.

We have shown how to compute the value $Z_{A,D}(G)$ in polynomial time by querying the oracle $\mathrm{EVAL}(A, D)$ on polynomially many instances G_n, for $n \ge 2$. It follows that $\mathrm{EVAL}(A, D) \le_{\mathrm{T}}^{\mathrm{P}} \mathrm{EVAL}_{\mathrm{simp}}(A, D)$. ◀

We are ready to prove the #P-hardness part of the dichotomy by Goldberg et al. [12] (Theorem 1.1) for simple graphs. Let A be a real symmetric $m \times m$ matrix. Assuming that A does not satisfy the tractability conditions of the dichotomy theorem of Goldberg et al., the problem $\mathrm{EVAL}(A)$ is #P-hard. By Theorem 11 (with $D = I_m$), $\mathrm{EVAL}(A) \le_{\mathrm{T}}^{\mathrm{P}} \mathrm{EVAL}_{\mathrm{simp}}(A)$. It follows that $\mathrm{EVAL}_{\mathrm{simp}}(A)$ is #P-hard.

Hence the dichotomy theorem by Goldberg et al. can improve to apply to simple graphs.

▶ **Theorem 12.** *Let A be a real symmetric matrix. Then either* $\mathrm{EVAL}(A)$ *is in polynomial time or* $\mathrm{EVAL}_{\mathrm{simp}}(A)$ *is #P-hard (a fortiori,* $\mathrm{EVAL}(A)$ *is #P-hard).*

Moreover, there is a polynomial time algorithm that, given the matrix A, decides which case of the dichotomy it is.

▶ Remark 13. The interpolation argument in Theorem 11 works even if G is a multigraph possibly with multiple loops at any vertex in the following sense. In Definition 4, we treat the loops of G as edges. We think of them as mapped to the entries A_{ii} in the evaluation of the partition function $Z_{A,D}$. However, we need to slightly change the way we define the graphs G_n. In addition to n-stretching the parallel edges of G, we also need to n-stretch each loop of G (i.e., replacing a loop by a closed path of length n). Now F is the set of parallel edges *and* loops in G. This way each $G_n = S_n^{(F)}(G)$ for $n \ge 2$ is simple and loopless. The rest of the proof goes through. In other words, the statement of Theorem 11 extends to a reduction from the $\mathrm{EVAL}(A, D)$ problem that allows input G to have multiloops, to the standard problem $\mathrm{EVAL}_{\mathrm{simp}}(A, D)$ not allowing loops.

References

1 Alexander I. Barvinok. *Combinatorics and Complexity of Partition Functions*, volume 30 of *Algorithms and combinatorics*. Springer, 2017.

2 Alexander I. Barvinok and Pablo Soberón. Computing the partition function for graph homomorphisms. *Combinatorica*, 37(4):633–650, 2017.

3 Rodney J. Baxter. The six and eight-vertex models revisited. *Journal of Statistical Physics*, 116(1):43–66, 2004.

4 Andrei Bulatov and Martin Grohe. The complexity of partition functions. *Theoretical Computer Science*, 348(2-3):148–186, 2005.

5 Jin-Yi Cai and Xi Chen. *Complexity Dichotomies for Counting Problems*, volume 1: Boolean Domain. Cambridge University Press, 2017. doi:10.1017/9781107477063.

6 Jin-Yi Cai and Xi Chen. A decidable dichotomy theorem on directed graph homomorphisms with non-negative weights. *Computational Complexity*, 28(3):345–408, 2019.

7 Jin-Yi Cai, Xi Chen, and Pinyan Lu. Graph homomorphisms with complex values: A dichotomy theorem. *SIAM Journal on Computing*, 42(3):924–1029, 2013.

8 Martin E. Dyer, Alan M. Frieze, and Mark Jerrum. On counting independent sets in sparse graphs. *SIAM Journal on Computing*, 31(5):1527–1541, 2002.

9 Martin E. Dyer, Leslie A. Goldberg, and Mike Paterson. On counting homomorphisms to directed acyclic graphs. *Journal of the ACM*, 54(6):27, 2007.

10 Martin E. Dyer and Catherine S. Greenhill. The complexity of counting graph homomorphisms. *Random Structures and Algorithms*, 17(3-4):260–289, 2000.

11 Martin E. Dyer and Catherine S. Greenhill. Corrigendum: The complexity of counting graph homomorphisms. *Random Structures and Algorithms*, 25(3):346–352, 2004.

12 Leslie A. Goldberg, Martin Grohe, Mark Jerrum, and Marc Thurley. A complexity dichotomy for partition functions with mixed signs. *SIAM Journal on Computing*, 39(7):3336–3402, 2010.

13 Martin Grohe and Marc Thurley. Counting homomorphisms and partition functions. In *Model Theoretic Methods in Finite Combinatorics*, volume 558 of *Contemporary Mathematics*, pages 243–292. American Mathematical Society, 2011.

14 Pavol Hell and Jaroslav Nešetřil. *Graphs and homomorphisms*, volume 28 of *Oxford lecture series in mathematics and its applications*. Oxford University Press, 2004.

15 Tyler Helmuth, Will Perkins, and Guus Regts. Algorithmic Pirogov-Sinai theory. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1009–1020, 2019.

16 Hendrik W. Lenstra Jr. Algorithms in algebraic number theory. *Bulletin of the American Mathematical Society*, 26(2):211–244, 1992.

17 Liang Li, Pinyan Lu, and Yitong Yin. Correlation decay up to uniqueness in spin systems. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 67–84, 2013.

18 László Lovász. Operations with structures. *Acta Mathematica Hungarica*, 18(3-4):321–328, 1967.

19 Han Peters and Guus Regts. Location of zeros for the partition function of the Ising model on bounded degree graphs. *arXiv preprint*, 2018. arXiv:1810.01699.

20 Alistair Sinclair, Piyush Srivastava, and Marc Thurley. Approximation algorithms for two-state anti-ferromagnetic spin systems on bounded degree graphs. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 941–953, 2012.

21 Allan Sly. Computational transition at the uniqueness threshold. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 287–296, 2010.

22 Marc Thurley. *The Complexity of Partition Functions*. PhD thesis, Humboldt Universität zu Berlin, 2009.

23 Marc Thurley. The complexity of partition functions on Hermitian matrices. *arXiv preprint*, 2010. arXiv:1004.0992.

24 Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

25 Dror Weitz. Counting independent sets up to the tree threshold. In *Proceedings of the 38th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 140–149, 2006.

26 Mingji Xia. Holographic reduction: A domain changed application and its partial converse theorems. *International Journal of Software and Informatics*, 5(4):567–577, 2011.