On the (In)tractability of Computing Normalizing Constants for the Product of Determinantal Point Processes

Naoto Ohsaka [1] **Tatsuya Matsuoka** [1]

Abstract

We consider the *product of determinantal point processes (DPPs)*, a point process whose probability mass is proportional to the product of principal minors of *multiple* matrices as a natural, promising generalization of DPPs. We study the computational complexity of computing its *normalizing constant*, which is among the most essential probabilistic inference tasks. Our complexity-theoretic results (almost) rule out the existence of efficient algorithms for this task, unless input matrices are forced to have favorable structures. In particular, we prove the following:
(1) Computing $\sum_S \det(\mathbf{A}_{S,S})^p$ exactly for every (fixed) *positive even integer* p is UP-hard and $\mathsf{Mod}_3\mathsf{P}$-hard, which gives a negative answer to an open question posed by Kulesza & Taskar (2012).
(2) $\sum_S \det(\mathbf{A}_{S,S}) \det(\mathbf{B}_{S,S}) \det(\mathbf{C}_{S,S})$ is NP-hard to approximate within a factor of $2^{\mathcal{O}(|\mathcal{I}|^{1-\epsilon})}$ for any $\epsilon > 0$, where $|\mathcal{I}|$ is the input size. This result is stronger than $\#\mathsf{P}$-hardness for the case of *two* matrices by Gillenwater (2014).
(3) There exists a $k^{\mathcal{O}(k)}|\mathcal{I}|^{\mathcal{O}(1)}$-time algorithm for computing $\sum_S \det(\mathbf{A}_{S,S}) \det(\mathbf{B}_{S,S})$, where k is "the maximum rank of $\mathbf{A}$ and $\mathbf{B}$" or "the treewidth of the graph induced by nonzero entries of $\mathbf{A}$ and $\mathbf{B}$." Such parameterized algorithms are said to be *fixed-parameter tractable*.

1. Introduction

Determinantal point processes (DPPs) offer an appealing probabilistic model to compactly express negative correlation among combinatorial objects (Macchi, 1975; Borodin & Rains, 2005). Given an $n \times n$ matrix $\mathbf{A}$, a DPP defines a probability measure on $2^{[n]}$ such that the probability of drawing subset $S \subseteq [n]$ is proportional to the principal minor $\det(\mathbf{A}_{S,S})$. Consider a subset selection task: given n items (e.g., images (Kulesza & Taskar, 2011)) associated with quality scores q_i and feature vectors $\boldsymbol{\phi}_i$ for each $i \in [n]$, we are asked to choose a small group of high-quality, diverse items. One can then construct $\mathbf{A}$ as $A_{i,j} = q_i q_j \boldsymbol{\phi}_i^\top \boldsymbol{\phi}_j$, resulting in that $\det(\mathbf{A}_{S,S})$ is the squared volume of the parallelepiped spanned by $\{q_i \boldsymbol{\phi}_i\}_{i \in S}$, which balances item quality and set diversity. Encouraged by efficient algorithms for many inference tasks such as normalization, sampling, and marginalization, DPPs have been applied to numerous machine learning tasks, e.g., image search (Kulesza & Taskar, 2011), video summarization (Gong et al., 2014), object retrieval (Affandi et al., 2014), sensor placement (Krause et al., 2008), and Nyström method (Li et al., 2016).

One of the recent research trends is to extend or generalize DPPs to express more complicated distributions. Computing the *normalizing constant* (a.k.a. partition function) for such new models is at the heart of efficient probabilistic inference. For example, we can efficiently sample a subset from partition DPPs (Celis et al., 2017), which are restricted to include a few elements from each group, by quickly evaluating their normalizing constant. Such tractability is, of course, not necessarily the case for every generalization.

In this paper, we consider a natural, (seemingly) promising generalization of DPPs involving *multiple* matrices. The *product DPP (*Π*-DPP)* given m matrices $\mathbf{A}^1, \ldots, \mathbf{A}^m$ defines the probability mass for each subset S as $\propto \det(\mathbf{A}^1_{S,S}) \cdots \det(\mathbf{A}^m_{S,S})$, which can be significantly expressive: It enables to *embed* some constraints to DPPs, e.g., those that are defined by partitions (Celis et al., 2017) and bipartite matching, and it contains *exponentiated DPPs* (Mariet et al., 2018) of an integer exponent as a special case. The computational complexity of its normalizing constant,

$$\sum_{S \subseteq [n]} \det(\mathbf{A}^1_{S,S}) \cdots \det(\mathbf{A}^m_{S,S}),$$

is almost nebulous, except for $m \leq 2$ (Gurvits, 2005; Gillenwater, 2014; Anari & Gharan, 2017) (see Section 1.2). Our research question is thus the following:

> *How hard (or easy) is it to compute normalizing constants for* Π*-DPPs?*

[1]NEC Corporation. Correspondence to: Naoto Ohsaka <ohsaka@nec.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

Table 1. Summary of complexity-theoretic results. Our results are colored in either red (negative) or blue (positive). $\mathcal{Z}_m(\mathbf{A}^1, \ldots, \mathbf{A}^m) = \sum_S \det(\mathbf{A}^1_{S,S}) \cdots \det(\mathbf{A}^m_{S,S})$ denotes the normalizing constant for Π-DPPs. n denotes the size of matrices, $|\mathcal{I}|$ is the number of bits required to represent $\mathbf{A}^1, \ldots, \mathbf{A}^m$, nz is the set of nonzero entries in a matrix, and tw is the treewidth (see Section 2).

exact / approx. / parameters	$\mathcal{Z}_p(\mathbf{A}, \ldots, \mathbf{A})$ exponentiated DPP	$\mathcal{Z}_2(\mathbf{A}, \mathbf{B})$	$\mathcal{Z}_3(\mathbf{A}, \mathbf{B}, \mathbf{C})$	$\mathcal{Z}_m(\mathbf{A}^1, \ldots, \mathbf{A}^m)$
exact	UP-hard & Mod_3P-hard (**Corollary 3.2**, $p = 2, 4, 6, \ldots$)	#P-hard (Gurvits, 2005; Gillenwater, 2014)		
approximation	e^n-approx. in polynomial time (Anari & Gharan, 2017)		$2^{\mathcal{O}(\lvert\mathcal{I}\rvert^{1-\epsilon})}$-approx. is NP-hard (**Theorem 4.1**)	
			1 is $2^{\mathcal{O}(\lvert\mathcal{I}\rvert^2)}$-approx. (**Observation 4.2**)	
m = number of matrices $r = \max_{i\in[m]} \mathrm{rank}(\mathbf{A}^i)$	(special case of $\to$)	FPT; $r^{\mathcal{O}(r)} n^{\mathcal{O}(1)}$ time (**Theorem 5.1**)	(special case of $\to$)	FPT; $r^{\mathcal{O}(mr)} n^{\mathcal{O}(1)}$ time (**Theorem 5.4**)
m = number of matrices $w = \mathrm{tw}(\bigcup_{i\in[m]} \mathrm{nz}(\mathbf{A}^i))$	(special case of $\to$)	FPT; $w^{\mathcal{O}(w)} n^{\mathcal{O}(1)}$-time (**Theorem 5.5**)	(special case of $\to$)	FPT; $w^{\mathcal{O}(mw)} n^{\mathcal{O}(1)}$-time (**Theorem 5.10**)
m = number of matrices $w = \max_{i\in[m]} \mathrm{tw}(\mathrm{nz}(\mathbf{A}^i))$	(same as $\uparrow$)	#P-hard ($w \le 3, m = 2$) (**Theorem 5.11**)		

1.1. Our Contributions

We present an intensive study on the computational complexity of computing normalizing constants for Π-DPPs. Our complexity-theoretic results summarized in Table 1 (almost) rule out the existence of efficient algorithms for this problem unless input matrices are forced to have favorable structures. Our quests can be partitioned into three types: intractability, inapproximability, and fixed-parameter tractability. We refer the reader to Appendix A for brief introduction of complexity classes. Paragraph headings are colored in either **red** (negative) or **blue** (positive).

Contribution 1: Intractability (Sect. 3)

We first analyze the hardness of computing normalizing constants *exactly*. Computing $\sum_S \det(\mathbf{A}_{S,S}) \det(\mathbf{B}_{S,S})$ for two matrices is #P-hard (Gillenwater, 2014). [1]

Exponentiated DPPs. Our first target is a special case where $\mathbf{A}^i = \mathbf{A}$ for all i, which includes *exponentiated DPPs* (Mariet et al., 2018) of an integer exponent. Computation of its normalizing constant $\sum_S \det(\mathbf{A}_{S,S})^p$ for $p > 0$ is originally motivated by the Hellinger distance between two DPPs (Kulesza & Taskar, 2012). We prove that for every (fixed) positive even integer $p = 2, 4, 6, \ldots$, it is UP-hard and Mod_3P-hard to compute this normalizing constant, even when $\mathbf{A}$ is a $(-1, 0, 1)$-matrix or a P-matrix (Corollary 3.2). Polynomial-time algorithms for it hence do not exist unless both IntegerFactorization $\in$ UP and GraphIsomorphism $\in \mathsf{Mod}_3$P are polynomial-time solvable. Our result gives a negative answer to an open question posed in Section 7.2 of (Kulesza & Taskar, 2012). We must emphasize that Gurvits (2005) already proved #P-hardness of computing $\sum_S \det(\mathbf{A}_{S,S})^2$ (see Section 1.2).

[1] #P is the class of function problems of counting the number of accepting paths of a nondeterministic polynomial-time Turing machine, and hence NP $\subseteq$ #P.

Contribution 2: Inapproximability (Sect. 4)

Understanding the hardness of exact computation, we then seek the possibility of *approximation*. Our hope is to guess an accurate estimate; e.g., an e^n-factor approximation is possible for two matrices (Anari & Gharan, 2017).

(Sub)exponential-Factor Inapproximability for $m = 3$. Unfortunately, hopes are dashed: We prove that it is NP-hard to approximate the normalizing constant for three matrices, i.e., $\sum_S \det(\mathbf{A}_{S,S}) \det(\mathbf{B}_{S,S}) \det(\mathbf{C}_{S,S})$, within a factor of $2^{\mathcal{O}(|\mathcal{I}|^{1-\epsilon})}$ or $2^{\mathcal{O}(n^{1/\epsilon})}$ for any $\epsilon > 0$ even when $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are positive semi-definite, where $|\mathcal{I}|$ is the input size (Theorem 4.1). For instance, even a $2^{n^{100}}$-approximation cannot be expected. Moreover, unless RP = NP, even *approximate sampling* is impossible; i.e., we cannot sample (in polynomial time) from a distribution whose total variation distance from the Π-DPP defined by $\mathbf{A}, \mathbf{B}, \mathbf{C}$ is at most $\frac{1}{3}$. The same hardness results hold for the case of four or more matrices (i.e., $m \ge 4$). On the other hand, a simple guess of the number 1 is proven to be a $2^{\mathcal{O}(|\mathcal{I}|^2)}$-factor approximation (Observation 4.2).

Connection to Mixed Discriminants for $m = 2$. We devise an equivalence between the normalizing constant for two matrices and *mixed discriminants*, which are #P-hard to compute (Observation 4.3). Currently, no fully polynomial-time randomized approximation scheme (FPRAS)[2] for mixed discriminants is known and its existence is believed to be false (Gurvits, 2005); hence, the Π-DPP for two matrices is unlikely to admit an FPRAS. Such implication is unexpected from Gillenwater (2014).

[2] An FPRAS is a randomized $(1+\epsilon)$-approximation algorithm that runs in time polynomial in the input size and ϵ^{-1} for $\epsilon > 0$.

Contribution 3: Fixed-Parameter Tractability (Sect. 5)

We finally resort to *parameterization*, which has succeeded in overcoming the difficulty of machine learning problems recently (Ganian et al., 2018; Eiben et al., 2019). Parameterized complexity (Downey & Fellows, 2012) is a research field aiming at classifying (typically, NP-hard) problems based on the computational complexity with respect to parameters. Given a *parameter* k independent of input size $|\mathcal{I}|$, a problem is *fixed-parameter tractable (FPT)* if it is solvable in time $f(k)|\mathcal{I}|^{\mathcal{O}(1)}$ for some computable function f. On the other hand, a problem solvable in time $|\mathcal{I}|^{f(k)}$ is *slice-wise polynomial (XP)*. While both have polynomial runtimes for every fixed k, the polynomial part is dramatically different between them ($|\mathcal{I}|^{\mathcal{O}(1)}$ or $|\mathcal{I}|^{f(k)}$). Selecting appropriate parameters is vital to devising the fixed-parameter tractability. We introduce three parameters, where the first two are FPT, and the third is unlikely to be even XP.

(1) Maximum Rank → FPT. *Rank* is a natural parameter for matrices. We can assume bounded-rank matrices for DPPs if feature vectors $\boldsymbol{\phi}_i$ are low-dimensional (Celis et al., 2018), or the largest possible subset is far smaller than the ground set size n; e.g., Gartrell et al. (2017) learned a matrix factorization of rank 15 while $n \approx 2{,}000$ from real-world data. We prove that there exists an $r^{\mathcal{O}(r)}n^{\mathcal{O}(1)}$-time FPT algorithm for computing the normalizing constant for two $n \times n$ positive semi-definite matrices $\mathbf{A}$ and $\mathbf{B}$, where r is the *maximum rank* of $\mathbf{A}$ and $\mathbf{B}$ (Theorem 5.1). The central idea is to decompose $\mathbf{A}$ and $\mathbf{B}$ into $n \times r$ rectangular matrices followed by the application of the Cauchy–Binet formula. Our parameterized algorithm can be generalized to the case for m matrices of rank at most r, increasing runtime to $r^{\mathcal{O}(mr)}n^{\mathcal{O}(1)}$ (Theorem 5.4).

(2) Treewidth of Union → FPT. *Treewidth* (Robertson & Seymour, 1986) is one of the most important graph-theoretic parameters, which measures the "tree-likeness" of a graph. Many hard problems on graphs are FPT when parameterized by treewidth (Cygan et al., 2015). Informally, the treewidth of a matrix is that of the graph induced by *nonzero entries* in the matrix; e.g., matrices of *bandwidth* b have treewidth $\mathcal{O}(b)$. If feature vectors $\boldsymbol{\phi}_i$ exhibit clustering properties (van der Maaten & Hinton, 2008), the similarity score $\boldsymbol{\phi}_i^\top \boldsymbol{\phi}_j$ between items from different clusters would be negligibly small, and such entries can be discarded to obtain a small-bandwidth matrix. We prove that there exists a $w^{\mathcal{O}(w)}n^{\mathcal{O}(1)}$-time FPT algorithm for computing the normalizing constant for two matrices $\mathbf{A}$ and $\mathbf{B}$, where w is the treewidth of the *union* of nonzero entries in $\mathbf{A}$ and $\mathbf{B}$ (Theorem 5.5). The proof is based on dynamic programming, which itself is typical but requires complicated procedures. Our algorithm can be generalized to the case of m matrices, increasing runtime to $w^{\mathcal{O}(mw)}n^{\mathcal{O}(1)}$ (Theorem 5.10).

(3) Maximum Treewidth → Unlikely to be XP. Our FPT algorithm in Theorem 5.5 implicitly benefits from the fact that two matrices $\mathbf{A}$ and $\mathbf{B}$ have nonzero entries in similar places. So, what happens if $\mathbf{A}$ and $\mathbf{B}$ are structurally different? Can we still get FPT when parameterized by the *maximum* treewidth of $\mathbf{A}$ and $\mathbf{B}$? Our answer is negative: Computing normalizing constants is #P-hard even if *both* $\mathbf{A}$ and $\mathbf{B}$ have treewidth at most 3 (Theorem 5.11), implying that even XP algorithms do not exist unless FP = #P (which is at least as strong as P = NP).

1.2. Related Work

Exponentiated DPPs (E-DPPs) of exponent $p > 0$ define the probability mass for subset S as $\propto \det(\mathbf{A}_{S,S})^p$. Markov chain Monte Carlo on E-DPPs for $p < 1$ is proven to mix rapidly as they are strongly log-concave (Anari et al., 2019; Robinson et al., 2019), implying an FPRAS for the normalizing constant. Mariet et al. (2018) investigate when a DPP defined by $\mathbf{A}^p$ is close to an E-DPP of exponent p for $\mathbf{A}$. Quite surprisingly, Gurvits (2005) has proven the #P-hardness of exactly computing $\sum_S \det(\mathbf{A}_{S,S})^2$ for a P-matrix $\mathbf{A}$ before Kulesza & Taskar (2012); Gillenwater (2014); however, this result seems to be not well-known in the machine learning community. Computing the normalizing constant for *two positive semi-definite* matrices is #P-hard as proven by Gillenwater (2014) but approximable within an e^n-factor in polynomial time (Anari & Gharan, 2017), which is an affirmative answer to an open question of Kulesza & Taskar (2012). Our study strengthens these results by showing the hardness for an E-DPP of exponent $p = 2, 4, 6, \ldots$, and the impossibility of exponential approximation and approximate sampling for *three* matrices.

Π-DPPs can be thought of as *log-submodular point processes* (Djolonga & Krause, 2014; Gotovos et al., 2015), whose probability mass for subset S is $\propto \exp(f(S))$, where f is a submodular set function.[3] Setting $f(S) = \log\det(\mathbf{A}^1_{S,S}) + \cdots + \log\det(\mathbf{A}^m_{S,S})$ coincides with Π-DPPs. Gotovos et al. (2015) devised a bound on the mixing time of a Gibbs sampler, though it is not helpful for our case because f can take $\log(0) = -\infty$ as a value.

Constrained DPPs output a subset S with probability $\propto \det(\mathbf{A}_{S,S})$ if S satisfies a specific constraint. Π-DPPs include partition-matroid constraints (Celis et al., 2017).

2. Preliminaries

For a positive integer n, let $[n] = \{1, 2, \ldots, n\}$ and $[0, n] = \{0, 1, 2, \ldots, n\}$. The imaginary unit is denoted $\mathrm{i} = \sqrt{-1}$. For a finite set S and an integer $k \in [0, |S|]$, we write $\binom{S}{k}$ for the family of all size-k subsets of S. For a statement P,

[3] We say that a set function $f : 2^{[n]} \to \mathbb{R}$ is *submodular* if $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$ for all $S, T \subseteq [n]$.

$[\![P]\!]$ is 1 if P is true, and 0 otherwise. The symmetric group on $[n]$, consisting of all permutations over $[n]$, is denoted $\mathfrak{S}_n$. We use $\sigma : S \rightarrowtail T$ for two same-sized sets S and T to mean a bijection from S to T, and $\sigma|_X$ for set X to denote the restriction of σ to $X \cap S$. The base of logarithms is 2.

We denote the $n \times n$ identity matrix by $\mathbf{I}_n$ and the $n \times n$ all-one matrix by $\mathbf{J}_n$. For an $m \times n$ matrix $\mathbf{A}$ and two subsets $S \subseteq [m], T \subseteq [n]$ of indices, we write $\mathbf{A}_S$ for the $|S| \times n$ submatrix whose rows are the rows of $\mathbf{A}$ indexed by S, and $\mathbf{A}_{S,T}$ for the $|S| \times |T|$ submatrix whose rows are the rows of $\mathbf{A}$ indexed by S and columns are the columns of $\mathbf{A}$ indexed by T. For a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, the *determinant* is $\det(\mathbf{A}) = \sum_{\sigma \in \mathfrak{S}_n} \operatorname{sgn}(\sigma) \prod_{i \in [n]} A_{i,\sigma(i)}$ and the *permanent* is $\operatorname{per}(\mathbf{A}) = \sum_{\sigma \in \mathfrak{S}_n} \prod_{i \in [n]} A_{i,\sigma(i)}$, where $\operatorname{sgn}(\sigma) = (-1)^{N(\sigma)}$ is the sign of a permutation $\sigma \in \mathfrak{S}_n$, and $N(\sigma)$ is the inversion number of σ. In particular, $\det(\mathbf{A}_{S,S})$ for any $S \subseteq [n]$ is called a *principal minor*. We define $\det(\mathbf{A}_{\emptyset,\emptyset}) = 1$. A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is called *positive semi-definite* if $\mathbf{x}^\top \mathbf{A} \mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$. A matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is called a *P-matrix* (resp. *P_0-matrix*) if its all principal minors are positive (resp. nonnegative). A positive semi-definite matrix is a P_0-matrix, but not vice versa. A real matrix $\mathbf{A}$ is a P-matrix whenever it has positive diagonal entries and is row diagonally dominant (i.e., $|A_{i,i}| > \sum_{j \neq i} |A_{i,j}|$ for all i). For a bijection σ from $S \subseteq [n]$ to $T \subseteq [n]$, we denote $\mathbf{A}(\sigma) = \prod_{i \in S} A_{i,\sigma(i)}$.

2.1. Determinantal Point Processes

Given a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, a *determinantal point process (DPP)* (Macchi, 1975; Borodin & Rains, 2005) defines a probability measure on the power set $2^{[n]}$ whose probability mass for $S \subseteq [n]$ is proportional to $\det(\mathbf{A}_{S,S})$.[4] Generally, a P_0-matrix is acceptable to define a probability distribution while positive semi-definite matrices are commonly-used (Gartrell et al., 2019). The normalizing constant for a DPP has a simple closed form: $\sum_{S \subseteq [n]} \det(\mathbf{A}_{S,S}) = \det(\mathbf{A}+\mathbf{I})$ (Kulesza & Taskar, 2012). This equality holds for any (not necessarily symmetric) real-valued matrix $\mathbf{A}$.

In this paper, we consider a point process whose probability mass is determined based on the *product* of principal minors for multiple matrices. Given m matrices $\mathbf{A}^1, \ldots, \mathbf{A}^m \in \mathbb{R}^{n \times n}$, *the product DPP ($\Pi$-DPP)* defines the probability mass for each subset $S \subseteq [n]$ as $\propto \det(\mathbf{A}^1_{S,S}) \cdots \det(\mathbf{A}^m_{S,S})$. We use $\mathcal{Z}_m(\mathbf{A}^1, \ldots, \mathbf{A}^m)$ to denote its *normalizing constant*; namely,

$$\mathcal{Z}_m(\mathbf{A}^1, \ldots, \mathbf{A}^m) = \sum_{S \subseteq [n]} \prod_{i \in [m]} \det(\mathbf{A}^i_{S,S}). \quad (1)$$

Since $\prod_{i \in [m]} \det(\mathbf{A}^i_{S,S})$ is easy to compute, evaluating $\mathcal{Z}_m$ is crucial for probability mass estimation. The objective in this paper is to elucidate the computational complexity of estimating $\mathcal{Z}_m$. We shall raise two examples of Π-DPPs.

Example 2.1 (Embedding partition and matching constraints)**.** Given a partition $\mathcal{P}$ of $[n]$, we can build $\mathbf{A}$ such that $\det(\mathbf{A}_{S,S}) = [\![S$ contains at most one element from each group of $\mathcal{P}]\!]$ by defining $A_{i,j} = [\![i, j$ belong to the same group$]\!]$. Given a bipartite graph whose edge set is $[n]$, we can build $\mathbf{A}$ and $\mathbf{B}$ such that $\det(\mathbf{A}_{S,S}) \det(\mathbf{B}_{S,S}) = [\![S$ has no common vertices$]\!]$ (Gillenwater, 2014); such S is called a matching.

Example 2.2 (Exponentiated DPPs)**.** Setting $\mathbf{A}^i = \mathbf{A}$ for all $i \in [m]$, we have the Π-DPP to be an exponentiated DPP of exponent $p = m \geq 1$, which sharpens the diversity nature of DPPs (Mariet et al., 2018).

2.2. Graph-Theoretic Concepts

We introduce several notions and definitions from graph theory, which play a crucial role in Section 5. Let $G = (V, E)$ be a graph, where V is a set of vertices, and E is a set of edges. We use (u, v) to denote an (undirected or directed) edge connecting u and v. We then define the treewidth of graphs and matrices. *Treewidth* (Halin, 1976; Robertson & Seymour, 1986; Arnborg & Proskurowski, 1989) is one of the most important notions in graph theory, which captures the "tree-likeness" of a graph (see Appendix E for example).

Definition 2.3. A *tree decomposition* of an undirected graph $G = (V, E)$ is a pair $(T, \{X_t\}_{t \in T})$, where T is a tree of which vertex $t \in T$, referred to as a *node*, is associated with a vertex subset $X_t \subseteq V$, referred to as a *bag*, such that **(1)** $\bigcup_{t \in T} X_t = V$, **(2)** for every edge $(u, v) \in E$, there exists node $t \in T$ such that $u, v \in X_t$, and **(3)** for every vertex $v \in V$, the set $T_u = \{t \mid v \in X_t\}$ induces a connected subtree of T. The *width* of a tree decomposition $(T, \{X_t\}_{t \in T})$ is defined as $\max_{t \in V} |X_t| - 1$. The *treewidth* of a graph G, denoted $\operatorname{tw}(G)$, is the minimum possible width among all tree decompositions of G.

For an $n \times n$ matrix $\mathbf{A}$, we denote $\operatorname{nz}(\mathbf{A}) = \{(i, j) \mid A_{i,j} \neq 0, i \neq j\}$. The treewidth of $\mathbf{A}$, denoted $\operatorname{tw}(\operatorname{nz}(\mathbf{A}))$, or $\operatorname{tw}(\mathbf{A})$, is defined as the treewidth of the graph $([n], \operatorname{nz}(\mathbf{A}))$. For example, $\operatorname{tw}(\mathbf{I}_n) = 1$, $\operatorname{tw}(\mathbf{J}_n) = n - 1$, and a matrix of bandwidth[5] b has treewidth $\mathcal{O}(b)$. One important property of tree decompositions is that any bag X_t is a *separator*: for three nodes t, t', t'' of T such that t is on the (unique) path from t' to t'', submatrices $\mathbf{A}_{X_{t'} \setminus X_t, X_{t''} \setminus X_t}$ and $\mathbf{A}_{X_{t''} \setminus X_t, X_{t'} \setminus X_t}$ must be zero-matrices.

Remark 2.4. Our algorithms parameterized by rank (Section 5.1) and by treewidth (Section 5.2) are incomparable in a sense that the identity matrix $\mathbf{I}_n$ has rank n and treewidth 1, and the all-one matrix $\mathbf{J}_n$ has rank 1 and treewidth $n - 1$.

[4] We adopt the L-ensemble form of Borodin & Rains (2005).

[5] The bandwidth of matrix $\mathbf{A}$ is defined as the smallest integer b such that $A_{i,j} = 0$ whenever $|i - j| > b$.

2.3. Computational Models

Since we use several *reductions* that transform an input for one problem to that for another problem, we introduce the notion of input size and computational model carefully.

The *size* of an input $\mathcal{I}$, denoted $|\mathcal{I}|$, is the number of bits required for representing $\mathcal{I}$. We assume that all the numbers appearing are *rational*. The size of a rational number $x = p/q \in \mathbb{Q}$ (where p and q are relatively prime integers) and a rational matrix $\mathbf{A} \in \mathbb{Q}^{m\times n}$ is defined as follows (Schrijver, 1999): $\mathrm{size}(x) = 1 + \lceil \log(|p|+1)\rceil + \lceil \log(|q|+1)\rceil$, and $\mathrm{size}(\mathbf{A}) = mn + \sum_{i\in[m],j\in[n]} \mathrm{size}(A_{i,j})$. The size of a graph is defined as the size of its incidence matrix.

Selection of computational models is crucial for determining the runtime of algorithms; e.g., multiplying two n-bit integers can be done in $\mathcal{O}(n \log n\, 8^{\log^* n})$ time on *Turing machines* (Harvey et al., 2016). Such a level of precision on Turing machines is not needed; for ease of analysis, we adopt the *unit-cost random access machine* model, which can perform basic arithmetic operations (e.g., add, subtract, multiply, and divide) in constant time. However, abusing unrealistically powerful models leads to an unreasonable conclusion: "iterating n times the operation $x \leftarrow x^2$, we can compute 2^{2^n}, a 2^n-bit integer, in $\mathcal{O}(n)$ time." To avoid such pitfalls, we will ensure that numbers produced during the execution of algorithms intermediately are of size $|\mathcal{I}|^{\mathcal{O}(1)}$.

3. Intractability of Exponentiated DPPs

We present the intractability of computing the normalizing constant for exponentiated DPPs of every positive even exponent, e.g., $\mathcal{Z}_4(\mathbf{A},\mathbf{A},\mathbf{A},\mathbf{A})$. For a positive number p and a matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, we denote $\mathcal{Z}^p(\mathbf{A}) = \sum_{S\subseteq[n]} \det(\mathbf{A}_{S,S})^p$. Our technical result is the following.

Theorem 3.1. *Computing $\mathcal{Z}^2(\mathbf{A}) \bmod 3$ for a matrix $\mathbf{A} \in \mathbb{Q}^{n\times n}$ is UP-hard and $\mathsf{Mod}_3\mathsf{P}$-hard. The same statement holds even when $\mathbf{A}$ is restricted to be either a $(-1,0,1)$-matrix or a P-matrix.*

As a corollary, we can show the same hardness for every *fixed* positive even integer p (i.e., p is not in the input), giving a negative answer to an open question of Kulesza & Taskar (2012), whose proof is deferred to Appendix B.

Corollary 3.2. *For every fixed positive even integer p, computing $\mathcal{Z}^p(\mathbf{A}) \bmod 3$ for either a $(-1,0,1)$-matrix or a P-matrix $\mathbf{A}$ is UP-hard and $\mathsf{Mod}_3\mathsf{P}$-hard.*

Our proof of Theorem 3.1 relies on the celebrated results relating $\mathcal{Z}^2$ to permanent by Kogan (1996), who presented an efficient algorithm for computing $\mathrm{per}(\mathbf{A}) \bmod 3$ for a matrix $\mathbf{A}$ with $\mathrm{rank}(\mathbf{A}\mathbf{A}^\top - \mathbf{I}_n) \le 1$. In the remainder of this subsection, *arithmetic operations are performed over modulo* 3, and the symbol $\equiv$ means congruence modulo 3.

Lemma 3.3 (Lemma 2.7 in (Kogan, 1996))**.** *Let $\mathbf{X}$ be a matrix such that $\det(\mathbf{X} + \mathrm{i}\mathbf{I}_n) \not\equiv 0$. Then, it holds that $\mathcal{Z}^2(\mathbf{X}) \equiv \det(\mathbf{X} + \mathrm{i}\mathbf{I}_n)^2 \,\mathrm{per}((\mathbf{I}_n + \mathrm{i}\mathbf{X})^{-1} + \mathbf{I}_n)$.*

Proof of Theorem 3.1. We reduce from a problem of computing the permanent of a $(0,1)$-matrix $\bmod 3$, which is UP-hard and $\mathsf{Mod}_3\mathsf{P}$-hard (Valiant, 1979) (cf. Theorem 2). Let $\mathbf{A}$ be an $n\times n$ $(0,1)$-matrix. By Proposition 2.2 in (Kogan, 1996), we compute a diagonal $(-1,1)$-matrix $\mathbf{D}$ in polynomial time such that $\mathbf{D}\mathbf{A} - \mathbf{I}_n$ is not singular and $\mathrm{per}(\mathbf{A}) \equiv \det(\mathbf{D})\,\mathrm{per}(\mathbf{D}\mathbf{A})$. We then compute $\mathbf{X} = \mathrm{i}^{-1}((\mathbf{D}\mathbf{A} - \mathbf{I}_n)^{-1} - \mathbf{I}_n)$ by Gaussian elimination modulo 3. Since $\det(\mathbf{X} + \mathrm{i}\mathbf{I}_n) \not\equiv 0$, we have by Lemma 3.3 that $\mathrm{per}(\mathbf{A}) \equiv \mathcal{Z}^2(\mathbf{X}) \det(\mathbf{D}) \det(\mathbf{X} + \mathrm{i}\mathbf{I}_n)^{-2}$. We transform $\mathbf{X}$ into $\mathbf{X}'$ according to the following two cases:

(1) $\det(\mathbf{D}) \det(\mathbf{X} + \mathrm{i}\mathbf{I}_n)^{-2} \equiv 1$: let $\mathbf{X}' = \mathbf{X}$.

(2) $\det(\mathbf{D}) \det(\mathbf{X} + \mathrm{i}\mathbf{I}_n)^{-2} \equiv 2$: let $\mathbf{X}' = \begin{bmatrix} \mathbf{X} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}^\top & \mathrm{i} & \mathrm{i} \\ \mathbf{0}^\top & \mathrm{i} & \mathrm{i} \end{bmatrix}$, where $\mathbf{0}$ is the $n\times 1$ zero-matrix. We have that $\mathcal{Z}^2(\mathbf{X}') \equiv 2\mathcal{Z}^2(\mathbf{X})$; remark that $\mathcal{Z}^2(\left[\begin{smallmatrix} \mathrm{i} & \mathrm{i} \\ \mathrm{i} & \mathrm{i} \end{smallmatrix}\right]) = -1$.

Consequently, we always have that $\mathrm{per}(\mathbf{A}) \equiv \mathcal{Z}^2(\mathbf{X}')$. Because entries of $\mathbf{X}'$ are purely imaginary numbers by construction, we can uniquely define a real-valued matrix $\mathbf{Y}$ such that $\mathbf{X}' = \mathrm{i}\mathbf{Y}$. Consider the polynomial $\mathcal{Z}^2(x\mathbf{Y})$ in x, i.e., $\sum_{S\subseteq[n]} x^{2|S|} \det(\mathbf{Y}_{S,S})^2 \equiv a_0 + a_1 x + a_2 x^2$ for some a_0, a_1, a_2. Solving a system of linear equations $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \equiv \begin{bmatrix} \mathcal{Z}^2(0\mathbf{Y}) \\ \mathcal{Z}^2(1\mathbf{Y}) \\ \mathcal{Z}^2(2\mathbf{Y}) \end{bmatrix}$ and noting that $\mathcal{Z}^2(1\mathbf{Y}) \equiv \mathcal{Z}^2(2\mathbf{Y})$, we have that $a_0 \equiv 1, a_1 \equiv 0, a_2 \equiv \mathcal{Z}^2(\mathbf{Y}) - 1$ and hence $\mathcal{Z}^2(\mathrm{i}\mathbf{Y}) \equiv 2 - \mathcal{Z}^2(\mathbf{Y})$. We transform $\mathbf{Y}$ into a $(-1,0,1)$-matrix $\mathbf{Y}'$ having the same permanent as $Y'_{i,j}$ is 0 if $Y_{i,j} \equiv 0$, $+1$ if $Y_{i,j} \equiv 1$, and -1 if $Y_{i,j} \equiv 2$. We further obtain a P-matrix as $\mathbf{Y}'' = \mathbf{Y}' + 3n\mathbf{I}_n$. Accordingly, deciding whether $\mathrm{per}(\mathbf{A}) \not\equiv 0$ is reduced to deciding whether $\mathcal{Z}^2(\mathbf{Y}') \not\equiv 2$ (and $\mathcal{Z}^2(\mathbf{Y}'') \not\equiv 2$), in polynomial time. □

4. Inapproximability for Three Matrices

Albeit the $\#\mathsf{P}$-hardness of $\mathcal{Z}_m$ for all $m \ge 2$, there is still a room to consider the approximability of $\mathcal{Z}_m$; e.g., Anari & Gharan (2017) have given an e^n-factor approximation algorithm for $\mathcal{Z}_2$. Unfortunately, we present a strong inapproximability for the case of $m \ge 3$.

4.1. (Sub)exponential-Factor Inapproximability

Our result shows a *subexponential-factor* inapproximation for the case of three matrices. We say that an estimate $\widehat{\mathcal{Z}}_m$ is a *ρ-approximation* to $\mathcal{Z}_m$ if $\rho^{-1}\mathcal{Z}_m \le \widehat{\mathcal{Z}}_m \le \rho\mathcal{Z}_m$. For two probability measures $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ on Ω, the *total variation distance* is defined as $\frac{1}{2}\sum_{x\in\Omega} |\mu_x - \eta_x|$. The proof reminds of that for the NP-hardness of three-matroid intersection.

Theorem 4.1. *For any fixed positive $\epsilon > 0$, it is NP-hard to approximate $\mathcal{Z}_3(\mathbf{A}, \mathbf{B}, \mathbf{C})$ for three matrices $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{Q}^{n \times n}$ within a factor of $2^{\mathcal{O}(|\mathcal{I}|^{1-\epsilon})}$ or $2^{\mathcal{O}(n^{1/\epsilon})}$, where $|\mathcal{I}|$ is the input size. Moreover, unless RP = NP, no polynomial-time algorithm can generate a random sample from a distribution whose total variation distance from the Π-DPP defined by $\mathbf{A}, \mathbf{B}, \mathbf{C}$ is at most $\frac{1}{3}$. The same statement holds if $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are restricted to be positive semi-definite.*

Proof. We show a polynomial-time reduction from an NP-complete HAMILTONIANPATH problem (Garey & Johnson, 1979), which, for a directed graph $G = (V, E)$ on n vertices and m edges, asks to find a directed simple path that visits every vertex of V exactly once (called a *Hamiltonian path*). Such G having a Hamiltonian path is called *Hamiltonian*.

We construct $m \times m$ three positive semi-definite matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ indexed by edges in E such that $\mathcal{Z}_3(\mathbf{A}, \mathbf{B}, \mathbf{C})$ is "significantly" large if G is Hamiltonian. We first define $\mathbf{A}$ and $\mathbf{B}$ so that $A_{i,j}$ is 1 if edges i, j share a common head and 0 otherwise, and $B_{i,j}$ is 1 if edges i, j share a common tail and 0 otherwise. For any $S \subseteq E$, $\det(\mathbf{A}_{S,S}) \det(\mathbf{B}_{S,S})$ takes 1 if S consists of directed paths or cycles only, and 0 otherwise. We then define $\mathbf{C}$ so that $\det(\mathbf{C}_{S,S}) = \Pr_T[S \subseteq T]$ for all $S \subseteq E$, where T is chosen from the uniform distribution over all spanning trees in (the undirected version of) G. Such $\mathbf{C}$ can be found in polynomial time: It holds that $\mathbf{C} = \mathbf{M}\mathbf{L}^\dagger\mathbf{M}^\top$ (Burton & Pemantle, 1993), where $\mathbf{M} \in \{-1, 0, 1\}^{m \times n}$ is the edge-vertex incidence matrix of G, and $\mathbf{L}^\dagger \in \mathbb{Q}^{n \times n}$ is the Moore-Penrose inverse of the Laplacian of G, which can be obtained as $(\mathbf{L} + \frac{1}{n}\mathbf{J}_n)^{-1} - \frac{1}{n}\mathbf{J}_n$ by Gaussian elimination in polynomial time (Edmonds, 1967; Schrijver, 1999). Since $m \leq n^2$, $\det(\mathbf{C}_{S,S})$ for $S \subseteq E$ is within the range between 2^{-n^2} and 1 if there exists a spanning tree T that includes S and 0 otherwise. It turns out that $\det(\mathbf{A}_{S,S}) \det(\mathbf{B}_{S,S}) \det(\mathbf{C}_{S,S})$ for $S \in \binom{E}{n-1}$ is positive *if and only if* S is a Hamiltonian path.

Redefine $\epsilon \leftarrow \lfloor 1/\epsilon \rfloor^{-1}$, which does not decrease the value of ϵ, and $\mathbf{A} \leftarrow \theta\mathbf{A}$, where $\theta = 2^{n^{4/\epsilon}} \in \mathbb{N}$. Since each entry of $\mathbf{A}$ and $\mathbf{B}$ is an integer at most θ and 1, respectively, $\mathrm{size}(\mathbf{A}) = \mathcal{O}(m^2 \log(2^{n^{4/\epsilon}})) = \mathcal{O}(n^{(4/\epsilon)+4})$ and $\mathrm{size}(\mathbf{B}) = \mathcal{O}(n^4)$. Since $\mathrm{size}(\mathbf{X}^{-1}) = \mathcal{O}(\mathrm{size}(\mathbf{X})n^2)$ for any $n \times n$ nonsingular matrix $\mathbf{X}$ (Schrijver, 1999) and $\mathrm{size}(\mathbf{L} + \frac{1}{n}\mathbf{J}) = \mathcal{O}(n^2 \log n)$, we have that $\mathrm{size}(\mathbf{L}^\dagger) = \mathcal{O}(n^4 \log n)$ and thus that $\mathrm{size}(\mathbf{C}) = m^2\mathcal{O}(n^4 \log n) = \mathcal{O}(n^8 \log n)$. Consequently, the input size is bounded by $|\mathcal{I}| = \mathcal{O}(n^{(4/\epsilon)+4}) + \mathcal{O}(n^4) + \mathcal{O}(n^8 \log n) = \mathcal{O}(n^{(4/\epsilon)+4})$, a polynomial in n (for fixed $\epsilon < 1$).

Now, we explain how to use $\mathcal{Z}_3$ to decide the Hamiltonicity of G. $\det(\mathbf{A}_{S,S}) \det(\mathbf{B}_{S,S}) \det(\mathbf{C}_{S,S})$ for edge set $S \subseteq E$ is 0 whenever "$|S| \geq n$," or "$|S| = n - 1$ but S is not Hamiltonian." Then, $\mathcal{Z}_3(\mathbf{A}, \mathbf{B}, \mathbf{C})$ can be decomposed into two sums $\sum_{S:|S|<n-1} \det(\mathbf{A}_{S,S}) \det(\mathbf{B}_{S,S}) \det(\mathbf{C}_{S,S}) + \sum_{S:\text{ Hamiltonian}} \det(\mathbf{A}_{S,S}) \det(\mathbf{B}_{S,S}) \det(\mathbf{C}_{S,S})$. We consider two cases.

(1) If there exists (at least) one Hamiltonian path S^* in G, then the sum is *at least* $\theta^{|S^*|}2^{-n^2} = 2^{n^{(4/\epsilon)+1} - n^{4/\epsilon} - n^2}$.

(2) If no Hamiltonian path exists in G, then, the sum is *at most* $\sum_{S:|S|<n-1} \theta^{n-2} \leq 2^{n^2}2^{n^{4/\epsilon}(n-2)} = 2^{n^{(4/\epsilon)+1} - 2n^{4/\epsilon} + n^2}$.

Hence, there is an exponential-factor gap $2^{n^{4/\epsilon} - 2n^2}$ between the two cases. Since $|\mathcal{I}|^{1-\epsilon} = \mathcal{O}(n^{(4/\epsilon)-4\epsilon})$, a $2^{\mathcal{O}(|\mathcal{I}|^{1-\epsilon})}$-factor or $2^{\mathcal{O}(n^{1/\epsilon})}$-factor approximation to $\mathcal{Z}_3$ suffices to distinguish the two cases (for sufficiently large n).

We then prove the second argument. Assume that G is Hamiltonian. Observe then that a random edge set drawn from the Π-DPP defined by $\mathbf{A}, \mathbf{B}, \mathbf{C}$ (denoted $\boldsymbol{\mu}$) is Hamiltonian with probability $\geq 1 - \frac{1}{1+2^{n^{4/\epsilon} - 2n^2}}$. Hence, provided a polynomial-time algorithm to generate random edge sets whose total variation distance from $\boldsymbol{\mu}$ is at most $\frac{1}{3}$, we can use it to verify the Hamiltonicity of G with probability $\geq \frac{2}{3} - \frac{1}{1+2^{n^{4/\epsilon} - 2n^2}} > \frac{1}{2}$ (whenever $n \geq 2$), implying that HAMILTONIANPATH $\in$ RP; hence, RP = NP. □

4.2. (Super)exponential-Factor Approximability

Whereas subexponential-factor approximation for $\mathcal{Z}_3$ is hard, we show a simple (super)exponential-factor approximation for $\mathcal{Z}_m$, whose proof is deferred to Appendix C.

Observation 4.2. *For m P_0-matrices $\mathbf{A}^1, \ldots, \mathbf{A}^m$, the number 1 is a $2^{\mathcal{O}(|\mathcal{I}|^2)}$-factor approximation to $\mathcal{Z}_m(\mathbf{A}^1, \ldots, \mathbf{A}^m)$, where $|\mathcal{I}|$ is the input size.*

4.3. Connection to Mixed Discriminants

We finally connect $\mathcal{Z}_2$ to mixed discriminants, which generalize the permanent. The *mixed discriminant* for m positive semi-definite matrices $\mathbf{K}^1 \ldots \mathbf{K}^m \in \mathbb{R}^{m \times m}$ is defined as $D(\mathbf{K}^1, \ldots, \mathbf{K}^m) = \frac{\partial^m}{\partial x_1 \cdots \partial x_m} \det(x_1\mathbf{K}^1 + \cdots + x_m\mathbf{K}^m)$.

We show below a polynomial-time reduction from mixed discriminants to $\mathcal{Z}_2$ as a corollary of (Celis et al., 2017), whose proof is deferred to Appendix C. Since the existence of an FPRAS for mixed discriminants is suspected to be false (Gurvits, 2005), our reduction implies that $\mathcal{Z}_2$ is unlikely to have an FPRAS. Such implication does not hold for Gillenwater (2014)'s reduction since they reduce from the problem of counting all matchings in a bipartite graph, which admits an FPRAS (Jerrum et al., 2004).

Observation 4.3. *Given m positive semi-definite matrices $\mathbf{K}^1, \ldots, \mathbf{K}^m \in \mathbb{Q}^{m \times m}$, we can construct two positive semi-definite matrices $\mathbf{A}, \mathbf{B} \in \mathbb{Q}^{m^2 \times m^2}$ such that the coefficient of x^m in polynomial $\mathcal{Z}_2(x\mathbf{A}, \mathbf{B})$ equals $m!\, D(\mathbf{K}^1, \ldots, \mathbf{K}^m)$, in polynomial time.*

5. Fixed-Parameter Tractability

In this section, we investigate the fixed-parameter tractability of computing $\mathcal{Z}_m$. Given a parameter k computable from input $\mathcal{I}$, a problem is said to be *fixed-parameter tractable (FPT)* and *slice-wise polynomial (XP)* if it is solvable in time $f(k)|\mathcal{I}|^{\mathcal{O}(1)}$ and $|\mathcal{I}|^{f(k)}$ for some computable function f, respectively. Our goal is either (1) to develop an FPT algorithm for an appropriate parameter, or (2) to disprove the existence of such algorithms under plausible assumptions.

5.1. Parameterization by Maximum Rank

We first consider the *maximum rank* of matrices as a parameter. Our theorem below demonstrates that computing $\mathcal{Z}_2(\mathbf{A}, \mathbf{B})$ for two positive semi-definite matrices $\mathbf{A}, \mathbf{B}$ parameterized by the maximum rank is FPT.

Theorem 5.1. *Let $\mathbf{A}, \mathbf{B}$ be two positive semi-definite matrices in $\mathbb{Q}^{n\times n}$ of rank at most r. Then, there exists an $r^{\mathcal{O}(r)}n^{\mathcal{O}(1)}$-time algorithm computing $\mathcal{Z}_2(\mathbf{A}, \mathbf{B})$ exactly.*

Before proceeding to the proof of Theorem 5.1, we introduce the following technical lemma, whose proof is based on dynamic programming and deferred to Appendix D.

Lemma 5.2. *Let $\mathbf{A}^1, \dots, \mathbf{A}^m$ be m matrices in $\mathbb{Q}^{n\times s}$, and $\sigma_1, \dots, \sigma_m \in \mathfrak{S}_s$ be m permutations over $[s]$. Then, $\sum_{S\subseteq\binom{[n]}{s}} \prod_{i\in[s]} (A^1_S)_{i,\sigma_1(i)} \cdot \dots \cdot (A^m_S)_{i,\sigma_m(i)}$ can be computed in time $\mathcal{O}(msn^2)$.*

We then introduce the Cauchy–Binet formula.

Lemma 5.3 (Cauchy–Binet formula)**.** *Let $\mathbf{A}$ be an $s \times r$ matrix and $\mathbf{B}$ be an $r \times s$ matrix. Then, the determinant of $\mathbf{AB}$ is $\det(\mathbf{AB}) = \sum_{C\in\binom{[r]}{s}} \det(\mathbf{A}_{[s],C}) \det(\mathbf{B}_{C,[s]})$.*

Proof of Theorem 5.1. We decompose $\mathbf{A}$ into two $n \times r$ rectangular matrices. For this purpose, we first compute an LDL decomposition[6] $\mathbf{A} = \mathbf{LDL}^\top$, where $\mathbf{L} \in \mathbb{Q}^{n\times n}$ and $\mathbf{D} \in \mathbb{Q}^{n\times n}$ is a diagonal matrix such that $D_{i,i} = 0$ for all $i \notin [r]$ (since the rank is at most r). This is always possible since $\mathbf{A}$ is positive semi-definite and can be done in polynomial time (O'Donnell & Ta, 2011). We further decompose $\mathbf{D}$ into the product of an $n \times r$ matrix $\mathbf{C}$ such that $C_{i,i} = D_{i,i}$ for all $i \in [r]$ and all the other elements are 0, and an $r \times n$ matrix $\mathbf{I}$ such that $I_{i,i} = 1$ for all $i \in [r]$ and all the other elements are 0. Setting $\mathbf{U} = \mathbf{LC} \in \mathbb{Q}^{n\times r}$ and $\mathbf{V} = \mathbf{LI}^\top \in \mathbb{Q}^{n\times r}$, we have that $\mathbf{A} = \mathbf{UV}^\top$. Similarly, we decompose $\mathbf{B} = \mathbf{XY}^\top$, where $\mathbf{X}, \mathbf{Y} \in \mathbb{Q}^{n\times r}$.

Because $\det(\mathbf{A}_{S,S}) \det(\mathbf{B}_{S,S}) = 0$ for all $S \subseteq [n]$ of size greater than r, we can expand $\mathcal{Z}_2(\mathbf{A}, \mathbf{B})$ using the Cauchy–Binet formula as follows.

$$\sum_{\substack{0\le s\le r\\ S\in\binom{[n]}{s}}} \sum_{C_1\in\binom{[r]}{s}} \det(\mathbf{U}_{S,C_1}\mathbf{V}^\top_{S,C_1}) \sum_{C_2\in\binom{[r]}{s}} \det(\mathbf{X}_{S,C_2}\mathbf{Y}^\top_{S,C_2}).$$

Observing that $|S| = |C_1| = |C_2|$, we further expand $\mathcal{Z}_2$ as

$$\sum_{\substack{0\le s\le r\\ C_1,C_2\in\binom{[r]}{s}}} \sum_{\sigma_1,\tau_1,\sigma_2,\tau_2\in\mathfrak{S}_s} \operatorname{sgn}(\sigma_1)\operatorname{sgn}(\tau_1)\operatorname{sgn}(\sigma_2)\operatorname{sgn}(\tau_2)\times \underbrace{\sum_{S\in\binom{[n]}{s}} \mathbf{U}_{S,C_1}(\sigma_1)\mathbf{V}_{S,C_1}(\tau_1)\mathbf{X}_{S,C_2}(\sigma_2)\mathbf{Y}_{S,C_2}(\tau_2)}_{\bigstar}.$$

Since $\bigstar$ can be evaluated in time $\mathcal{O}(sn^2)$ by Lemma 5.2, we can take the sum of $\Diamond$ over all possible combinations of $s, C_1, C_2, \sigma_1, \tau_1, \sigma_2, \tau_2$ in time $\sum_{0\le s\le r}\binom{r}{s}^2(s!)^4\mathcal{O}(sn^2) = \mathcal{O}(r^{4r}r^2n^2)$. Consequently, the overall computation time is bounded by $r^{\mathcal{O}(r)}n^2$. □

Theorem 5.1 can be generalized to the case of m matrices $\mathbf{A}^1, \dots, \mathbf{A}^m$; computation of $\mathcal{Z}_m$ parameterized by the maximum rank $\max_{i\in[m]} \operatorname{rank}(\mathbf{A}^i)$ *plus* the number of matrices m is FPT, whose proof is deferred to Appendix D.

Theorem 5.4. *For a positive integer m, let $\mathbf{A}^1, \dots, \mathbf{A}^m$ be m positive semi-definite matrices in $\mathbb{Q}^{n\times n}$ of rank at most r. Then, there exists an $r^{\mathcal{O}(mr)}n^{\mathcal{O}(1)}$-time algorithm computing $\mathcal{Z}_m(\mathbf{A}^1, \dots, \mathbf{A}^m)$ exactly.*

5.2. Parameterization by Treewidth of Union

We then consider the *treewidth* of the graph induced by the *union* of nonzero entries as a parameter. Our technical result below demonstrates that computing $\mathcal{Z}_2(\mathbf{A}, \mathbf{B})$ parameterized by $\operatorname{tw}(\operatorname{nz}(\mathbf{A}) \cup \operatorname{nz}(\mathbf{B}))$ is FPT.

Theorem 5.5. *Let $\mathbf{A}, \mathbf{B}$ be two matrices in $\mathbb{Q}^{n\times n}$. Then, there exists a $w^{\mathcal{O}(w)}n^{\mathcal{O}(1)}$-time algorithm that, given a tree decomposition of the graph $([n], \operatorname{nz}(\mathbf{A}) \cup \operatorname{nz}(\mathbf{B}))$ of width at most w, computes $\mathcal{Z}_2(\mathbf{A}, \mathbf{B})$ exactly.*

Remark 5.6. To construct "reasonable" tree decompositions, we can use existing algorithms, e.g., a $2^{\mathcal{O}(w)}n$-time 5-approximation algorithm by Bodlaender, Drange, Dregi, Fomin, Lokshtanov, and Pilipczuk (2016).

Proof Strategy. Our proof is based on dynamic programming upon tree decompositions. We first assume to be given a *nice tree decomposition* $(T, \{X_t\}_{t\in T})$, a convenient form of tree decompositions (see Appendix E for formal definition). Think of T as a rooted tree by referring to a fixed vertex r as the *root* of T, which naturally introduces notions of parents, child, and leaves. One useful property of nice

[6] We do not adopt the Cholesky decomposition to avoid square root computation.

tree decompositions is that $X_r = \emptyset$ and $X_\ell = \emptyset$ for every leaf ℓ of T. We denote $V_t = \bigcup_{t' \text{ in subtree rooted at } t} X_{t'}$; in particular, $V_r = [n]$.

Next we design dynamic programming tables. Given a nice tree decomposition $(T, \{X_t\}_{t\in T})$ of the graph $([n], \mathrm{nz}(\mathbf{A}) \cup \mathrm{nz}(\mathbf{B}))$, we aim to compute the following for each node t:

$$\sum_{\substack{S\subseteq V_t\setminus X_t \\ O_{A1},O_{A2}\subseteq X_t:|O_{A1}|=|O_{A2}|,\sigma_A:S\cup O_{A1}\rightarrowtail S\cup O_{A2} \\ O_{B1},O_{B2}\subseteq X_t:|O_{B1}|=|O_{B2}|,\sigma_B:S\cup O_{B1}\rightarrowtail S\cup O_{B2}}} \mathrm{sgn}(\sigma_A)\,\mathrm{sgn}(\sigma_B)\mathbf{A}(\sigma_A)\mathbf{B}(\sigma_B). \tag{2}$$

In particular, Eq. (2) is $\mathcal{Z}_2(\mathbf{A}, \mathbf{B})$ at the root r since $X_r = \emptyset, V_r = [n]$. We then discuss how to group exponentially many bijections. *States for node t* are defined as a tuple $\mathtt{st} = (O_1, O_2, F_1, F_2, \tau, N)$ such that $O_1, O_2 \subseteq X_t$ with $|O_1| = |O_2|$, $F_1 \subseteq O_1, F_2 \subseteq O_2$ with $|F_1| = |F_2|$, $\tau : O_1 \setminus F_1 \rightarrowtail O_2 \setminus F_2$, and $N \in [0, n^2]$.

We say that a bijection σ is *consistent* with $S \subseteq V_t \setminus X_t$ and $\mathtt{st} = (O_1, O_2, F_1, F_2, \tau, N)$ if **(1)** σ is a bijection $S \cup O_1 \rightarrowtail S \cup O_2$, **(2)** $F_1 = \sigma^{-1}(S) \cap O_1, F_2 = \sigma(S) \cap O_2$, **(3)** $\tau = \sigma|_{O_1\setminus F_1}$, and **(4)** $N = N(\sigma)$. We first show that any bijection σ appearing in Eq. (2) is consistent with a unique pair of S and $\mathtt{st}$, whose proof is deferred to Appendix E.

Lemma 5.7. *For a node t and subsets $S \subseteq V_t \setminus X_t, O_1, O_2 \subseteq X_t$ with $|O_1| = |O_2|$, let $\sigma : S \cup O_1 \rightarrowtail S \cup O_2$ be a bijection. Then, there exists a unique pair of $S \subseteq V_t \setminus X_t$ and $\mathtt{st}$ for t that σ is consistent with.*

We will use $\sigma(S, \mathtt{st})$ to denote the set of all bijections σ consistent with S and $\mathtt{st}$. By Lemma 5.7, we have that the collection of $\sigma(S, \mathtt{st})$ forms a partition of the set $\{\sigma : S \cup O_1 \rightarrowtail S \cup O_2 \mid S \subseteq V_t \setminus X_t, O_1 \subseteq X_t, O_2 \subseteq X_t, |O_1| = |O_2|\}$. We can then express Eq. (2) as follows:

$$\sum_{\substack{\mathtt{st}_A \text{ for } t \\ \mathtt{st}_B \text{ for } t \\ s\in[0,n]}} (-1)^{N_A+N_B} \sum_{S\subseteq\binom{V_t\setminus X_t}{s}} \Upsilon_{t,A}(S, \mathtt{st}_A) \cdot \Upsilon_{t,B}(S, \mathtt{st}_B),$$

where $\Upsilon_{t,A}(S, \mathtt{st}_A) = \sum_{\sigma_A\in\sigma(S,\mathtt{st}_A)} \mathbf{A}(\sigma_A)$ and $\Upsilon_{t,B}(S, \mathtt{st}_B) = \sum_{\sigma_B\in\sigma(S,\mathtt{st}_B)} \mathbf{B}(\sigma_B)$.

We define the table $dp_{t,s}$ for each $t \in T$ and $s \in [0, n]$ to store the following quantity with key $\begin{bmatrix}\mathtt{st}_A \\ \mathtt{st}_B\end{bmatrix}$:

$$dp_{t,s}\begin{bmatrix}\mathtt{st}_A \\ \mathtt{st}_B\end{bmatrix} = \sum_{S\in\binom{V_t\setminus X_t}{s}} \Upsilon_{t,A}(S, \mathtt{st}_A) \cdot \Upsilon_{t,B}(S, \mathtt{st}_B).$$

By definition, the number of possible states is at most $2^{|X_t|}2^{|X_t|}2^{|X_t|}2^{|X_t|}|X_t|!(n^2+1) \le 16^{w+1}(w+1)!(n^2+1)$; $dp_{t,s}$ contains $w^{\mathcal{O}(w)}n^{\mathcal{O}(1)}$ entries. We are now ready to construct $dp_{t,s}$ given already-filled $dp_{t',s'}$ for children t' of t and $s' \in [0, n]$, whose proof is deferred to Appendix E.

Lemma 5.8. *Let t be a non-leaf node, and $s \in [0, n]$. Given $dp_{t',s'}$ for all children t' of t and $s' \in [0, n]$, we can compute each entry of $dp_{t,s}$ in time $w^{\mathcal{O}(w)}n^{\mathcal{O}(1)}$.*

Proof of Theorem 5.5. Our parameterized algorithm works as follows. Given a tree decomposition for $([n], \mathrm{nz}(\mathbf{A}) \cup \mathrm{nz}(\mathbf{B}))$ of width at most w, we transform it to a nice tree decomposition $(T, \{X_t\}_{t\in T})$ rooted at r of width at most w that has $\mathcal{O}(wn)$ nodes in polynomial time (Cygan et al., 2015). For every leaf ℓ of T, we initialize $dp_{\ell,s}$ so that $dp_{\ell,s}\left[\begin{smallmatrix}\emptyset,\emptyset,\emptyset,\emptyset,\emptyset\rightarrowtail\emptyset,0 \\ \emptyset,\emptyset,\emptyset,\emptyset,\emptyset\rightarrowtail\emptyset,0\end{smallmatrix}\right] = [\![s = 0]\!]$. Then, for each non-leaf node $t \in T$, we apply Lemma 5.8 to fill $dp_{t,s}$ using already-filled $dp_{t',s'}$ for t's children t' in a bottom-up fashion. Completing dynamic programming, we compute $\mathcal{Z}_2$ as $\sum_{s,N_A,N_B}(-1)^{N_A+N_B}dp_{r,s}\left[\begin{smallmatrix}\emptyset,\emptyset,\emptyset,\emptyset,\emptyset\rightarrowtail\emptyset,N_A \\ \emptyset,\emptyset,\emptyset,\emptyset,\emptyset\rightarrowtail\emptyset,N_B\end{smallmatrix}\right]$. The correctness follows from Lemmas 5.7 and 5.8. Because each table is of size $w^{\mathcal{O}(w)}n^{\mathcal{O}(1)}$, and each table entry can be computed in time $w^{\mathcal{O}(w)}n^{\mathcal{O}(1)}$ by Lemma 5.8, the whole time complexity is bounded by $w^{\mathcal{O}(w)}n^{\mathcal{O}(1)}$. □

Remark 5.9. Our dynamic programming implies the fixed-parameter tractability for *permanental processes* (Macchi, 1975) since $\sum_{S\subseteq[n]}\mathrm{per}(\mathbf{A}_S)\,\mathrm{per}(\mathbf{B}_S) = \sum_{s,N_A,N_B} dp_{r,s}\left[\begin{smallmatrix}\emptyset,\emptyset,\emptyset,\emptyset,\emptyset\rightarrowtail\emptyset,N_A \\ \emptyset,\emptyset,\emptyset,\emptyset,\emptyset\rightarrowtail\emptyset,N_B\end{smallmatrix}\right]$.

Theorem 5.5 can be generalized to the case of m matrices $\mathbf{A}^1, \dots \mathbf{A}^m$. Computing $\mathcal{Z}_m$ parameterized by the treewidth of $\mathrm{nz}(\mathbf{A}^1) \cup \cdots \cup \mathrm{nz}(\mathbf{A}^m)$ *plus* the number of matrices m is FPT, whose proof is deferred to Appendix E.

Theorem 5.10. *For a positive integer m, let $\mathbf{A}^1, \dots, \mathbf{A}^m$ be m matrices in $\mathbb{Q}^{n\times n}$. Then, there exists a $w^{\mathcal{O}(mw)}n^{\mathcal{O}(1)}$-time algorithm that, given a tree decomposition of the graph $([n], \bigcup_{i\in[m]}\mathrm{nz}(\mathbf{A}^i))$ of width at most w, computes $\mathcal{Z}_m(\mathbf{A}^1, \dots, \mathbf{A}^m)$ exactly.*

5.3. Parameterization by Maximum Treewidth

Let us finally take the *maximum treewidth* of two matrices as a parameter, and we refute its fixed-parameter tractability. This parameterization is preferable to the previous one because the maximum treewidth can be far smaller than the treewidth of the union. One can, for example, construct two $n \times n$ matrices $\mathbf{A}, \mathbf{B}$ such that $\max\{\mathrm{tw}(\mathbf{A}), \mathrm{tw}(\mathbf{B})\} = 1$ and $\mathrm{tw}(\mathrm{nz}(\mathbf{A}) \cup \mathrm{nz}(\mathbf{B})) = \mathcal{O}(\sqrt{n})$ because we can "weave" two paths into a grid. Unluckily, even when both of two matrices $\mathbf{A}, \mathbf{B}$ have treewidth at most 3, it is still #P-hard to compute $\mathcal{Z}_2(\mathbf{A}, \mathbf{B})$. Parameterization by $\max\{\mathrm{tw}(\mathbf{A}), \mathrm{tw}(\mathbf{B})\}$ is thus not even XP unless FP = #P. The proof is based on (Vadhan, 2001; Gillenwater, 2014) and deferred to Appendix F.

Theorem 5.11. *Let $\mathbf{A}, \mathbf{B}$ be two positive semi-definite $(0, 1)$-matrices such that $\mathrm{tw}(\mathbf{A}) \le 3$ and $\mathrm{tw}(\mathbf{B}) \le 3$. Then, computing $\mathcal{Z}_2(\mathbf{A}, \mathbf{B})$ is #P-hard.*

6. Concluding Remarks and Open Questions

In this paper, we studied the computational complexity of normalizing product determinantal point processes. Our results (almost) ruled out the possibility of efficient algorithms for general cases and devised the fixed-parameter tractability. Several immediate open questions are listed below.

- Can we prove the intractability of computing $\mathcal{Z}^p$ for p which is not a positive even integer, such as $p = 3$ and $p = 1.1$? Our proof strategy would no longer work.
- Can we prove the inapproximability of estimating $\mathcal{Z}^p$, or develop approximation algorithms for it? The current best upper bound for $p = 2$ is e^n due to Anari & Gharan (2017).
- Can we develop more "practical" FPT algorithms having a small exponential factor, e.g., $f(k) = 2^k$? We should avoid enumerating permutations.
- Can we establish fixed-parameter tractability for other parameters?

Acknowledgements

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions, and acknowledge helpful conversations with Shinji Ito.

Appendix

A. Brief Introduction to Complexity Classes

Decision Problems.

- P: The class of decision problems solvable by a deterministic polynomial-time Turing machine. Examples include LINEARPROGRAMMING and PRIMES (Q. Is an input integer prime?).
- NP: The class of decision problems solvable by a nondeterministic polynomial-time Turing machine (NP machine). Examples include SAT (Q. Is there a truth assignment satisfying an input Boolean formula?). It is widely believed that $\mathsf{P} \neq \mathsf{NP}$.
- UP: The class of decision problems solvable by an NP machine with *at most one* accepting path. Note that $\mathsf{P} \subseteq \mathsf{UP} \subseteq \mathsf{NP}$, but it is unknown if the inclusion is strict. Examples include UNAMBIGUOUSSAT (Q. Is there a truth assignment satisfying an input Boolean formula that is restricted to have at most one satisfying assignment?) and INTEGERFACTORIZATION (Q. Is there a factor $d \in [m]$ of an integer n given n and m?), for which no polynomial-time algorithms are known.
- RP: The class of decision problems for which a probabilistic polynomial-time Turing machine exists such that (1) if the answer is "yes", then it returns "yes" with probability at least $1/2$, and (2) if the answer is "no", then it always returns "no". Note that $\mathsf{P} \subseteq \mathsf{RP} \subseteq \mathsf{NP}$, and it is believed that $\mathsf{RP} \neq \mathsf{NP}$.

Counting-Related Problems.

- FP: The class of functions computable by a deterministic polynomial-time Turing machine, which is a function problem analogue of P. Examples include DETERMINANT (Q. Compute the determinant of an input square matrix), which is efficiently-computable via Gaussian elimination.
- $\#\mathsf{P}$: The class of function problems of counting the number of accepting paths of an NP machine. Examples of $\#\mathsf{P}$-complete problems include PERMANENT (Q. Compute the permanent of an input matrix) and $\#$SAT (Q. Counting the number of truth assignments satisfying an input Boolean formula). Note that $\mathsf{P} \neq \mathsf{NP}$ implies $\mathsf{FP} \neq \#\mathsf{P}$.
- $\mathsf{Mod}_k\mathsf{P}$: The class of decision problems solvable by an NP machine, where the number of accepting paths *is not divisible by* k. Examples include GRAPHISOMORPHISM (Q. Is there an "edge-preserving" bijection between the vertex sets of two input graphs), which is in $\mathsf{Mod}_k\mathsf{P}$ for all k (Arvind & Kurur, 2006).

Parameterized Problems.

- FPT (fixed-parameter tractable): The class of problems (with parameter k) solvable in time $f(k)|\mathcal{I}|^{\mathcal{O}(1)}$ for some computable function f. Examples include k-VERTEXCOVER (Q. Is there a k-vertex set including at least one endpoint of every edge of an n-vertex graph?), for which an $\mathcal{O}(1.2738^k + kn)$-time algorithm is known (Chen et al., 2010).
- XP (slice-wise polynomial): The class of problems (with parameter k) solvable in time $|\mathcal{I}|^{f(k)}$ for some computable function f. Examples include k-CLIQUE (Q. Is there a size-k complete subgraph in an n-vertex graph?), for which a brute-force search algorithm runs in $n^{\mathcal{O}(k)}$ time. k-CLIQUE is believed to be not in FPT; $\mathsf{FPT} \neq \mathsf{XP}$ is a plausible assumption in parameterized complexity.

B. Proof in Section 3

Proof of Corollary 3.2. Since $0^p \equiv 0^2, 1^p \equiv 1^2, 2^p \equiv 2^2 \mod 3$ if p is a positive even integer, we have that $\mathcal{Z}^p(\mathbf{A}) \equiv \mathcal{Z}^2(\mathbf{A}) \mod 3$. □

C. Proofs in Section 4

Proof of Observation 4.2. Let $\mathbf{A}^1, \ldots, \mathbf{A}^m$ be m P_0-matrices in $\mathbb{Q}^{n\times n}$, and $|\mathcal{I}| = \sum_{i\in[m]} \mathrm{size}(\mathbf{A}^i)$ be the input size. Of course, $\mathcal{Z}_m$ is bounded from below by 1; we shall show an upper bound. Applying Hadamard's inequality, we have that all principal minors are at most $M^n n^{n/2}$, where M is the maximum absolute entry in the m matrices. Hence,

$$\sum_{S\subseteq[n]} \det(\mathbf{A}^1_{S,S}) \cdots \det(\mathbf{A}^m_{S,S}) \leq 2^n (M^n n^{n/2})^m = 2^{n+mn\log M + \frac{mn}{2}\log n} = 2^{\mathcal{O}(|\mathcal{I}|^2)},$$

where the last deformation comes from that $|\mathcal{I}| \geq \log M$ and $|\mathcal{I}| \geq mn^2$. Thus, $\mathcal{Z}_m$ is between 1 and $2^{\mathcal{O}(|\mathcal{I}|^2)}$. □

Proof of Observation 4.3. Let $n = m^2$. Given m positive semi-definite matrices $\mathbf{K}^1, \dots, \mathbf{K}^m \in \mathbb{Q}^{m \times m}$, according to (Celis et al., 2017) (cf. Proof of Lemma 12), we construct an $n \times n$ positive semi-definite matrix $\mathbf{A}$ and an equal-sized partition $\mathcal{P}$ of $[n]$ such that $m!\, D(\mathbf{K}^1, \dots, \mathbf{K}^m) = \sum_{S \in \mathcal{C}} \det(\mathbf{A}_{S,S})$, where $\mathcal{C} = \{S \in \binom{[n]}{m} \mid |S \cap P| = 1 \ \forall P \in \mathcal{P}\}$. We then construct an $n \times n$ positive semi-definite matrix $\mathbf{B}$ such that $B_{i,j} = [\![i, j \text{ belong to the same group of } \mathcal{P}]\!]$; it turns out that $\sum_{S \in \binom{[n]}{m}} \det(\mathbf{A}_{S,S}) \det(\mathbf{B}_{S,S}) = \sum_{S \in \mathcal{C}} \det(\mathbf{A}_{S,S})$. Consider a polynomial $\mathcal{Z}_2(x\mathbf{A}, \mathbf{B})$ of degree n in $x \in \mathbb{Q}$, whose coefficient on x^m is exactly equal to the desired value, which completes the proof. Note that we can recover this by evaluating $\mathcal{Z}_2(x\mathbf{A}, \mathbf{B})$ for $n+1$ distinct values of x (say, $[n+1]$) followed by the application of Lagrangian interpolation. □

D. Proofs in Section 5.1

Proof of Lemma 5.2. The proof is based on dynamic programming. We first define a table dp of size $s \times n$, whose entry for each $\ell \in [s], o \in [n]$ is defined as

$$dp[\ell, o] = \sum_{\substack{S \subseteq \binom{[n]}{\ell} \\ \max(S) = o}} \prod_{i \in [\ell]} (A^1_S)_{i,\sigma_1(i)} \cdots (A^m_S)_{i,\sigma_m(i)}.$$

The desired value is equal to $\sum_{o \in [n]} dp[s, o]$. Observe then that for $\ell \in \{2, \dots, s\}, o \in [n]$

$$dp[\ell, o] = \sum_{1 \le o' < o} dp[\ell - 1, o'] A^1_{o,\sigma_1(\ell)} \cdots A^m_{o,\sigma_m(\ell)},$$
$$dp[1, o] = A^1_{o,\sigma_1(1)} \cdots A^m_{o,\sigma_m(1)}.$$

Note that the number of bits required to express each entry is bounded by $\log(2^n)(\mathrm{size}(\mathbf{A}^1) + \cdots + \mathrm{size}(\mathbf{A}^m))$, which is a polynomial in the input size. Calculating $dp[\ell, o]$ given $dp[\ell - 1, o']$ for all $o' \in [n]$ by $\mathcal{O}(nm)$ arithmetic operations, standard dynamic programming fills all entries of dp by $\mathcal{O}(msn^2)$ arithmetic operations. □

Proof of Theorem 5.4. We first decompose each of the m matrices into the product of two $n \times r$ rectangular matrices, i.e., $\mathbf{A}^i = \mathbf{X}^i (\mathbf{Y}^i)^\top$ for all $i \in [m]$, by LDL decomposition. Similarly to the proof of Theorem 5.1, we can expand $\mathcal{Z}_m(\mathbf{A}^1, \dots, \mathbf{A}^m)$ as follows.

$$\begin{aligned}
&\sum_{S \subseteq [n]} \det(\mathbf{A}^1_{S,S}) \cdots \det(\mathbf{A}^m_{S,S}) \\
&= \sum_{0 \le s \le r} \sum_{S \in \binom{[n]}{s}} \sum_{C_1 \in \binom{[r]}{s}} \det(\mathbf{X}^1_{S,C_1}) \det((\mathbf{Y}^1_{S,C_1})^\top) \cdots \sum_{C_m \in \binom{[r]}{s}} \det(\mathbf{X}^m_{S,C_m}) \det((\mathbf{Y}^m_{S,C_m})^\top) \\
&= \sum_{0 \le s \le r} \sum_{\substack{C_1 \in \binom{[r]}{s} \\ \sigma_1, \tau_1 \in \mathfrak{S}_s}} \cdots \sum_{\substack{C_m \in \binom{[r]}{s} \\ \sigma_m, \tau_m \in \mathfrak{S}_s}} \mathrm{sgn}(\sigma_1) \mathrm{sgn}(\tau_1) \cdots \mathrm{sgn}(\sigma_m) \mathrm{sgn}(\tau_m) \times \\
&\qquad \underbrace{\sum_{S \in \binom{[n]}{s}} \sum_{i \in [s]} (X^1_{S,C_1})_{i,\sigma_1(i)} (Y^1_{S,C_1})_{i,\tau_1(i)} \cdots (X^m_{S,C_m})_{i,\sigma_m(i)} (Y^m_{S,C_m})_{i,\tau_m(i)}}_{\clubsuit}.
\end{aligned}$$

By applying Lemma 5.2, we can calculate ♣ in time $\mathcal{O}(msn^2)$, and thus the entire computation time is bounded by

$$\sum_{0 \le s \le r} \binom{r}{s}^m (s!)^{2m} \mathcal{O}(msn^2) = \mathcal{O}(r^{2mr} r^2 n^2) = r^{\mathcal{O}(mr)} n^2.$$

□

E. Proofs in Section 5.2

We first define nice tree decompositions formally. Figures 1–4 show an example of (nice) tree decompositions.

Definition E.1 (Nice tree decomposition)**.** A tree decomposition $(T, \{X_t\}_{t\in T})$ rooted at r is called *nice* if

(1) every leaf and the root have empty bags; i.e., $X_r = \emptyset$ and $X_\ell = \emptyset$ for every leaf ℓ of T, and

(2) every non-leaf node is one of the following:

1. **Introduce node:** a node t with exactly one child t' such that $X_t = X_{t'} \cup \{v\}$ for some $v \notin X_{t'}$.
2. **Forget node:** a node t with exactly one child t' such that $X_t = X_{t'} \setminus \{v\}$ for some $v \in X_{t'}$.
3. **Join node:** a node t with exactly two children t', t'' such that $X_t = X_{t'} = X_{t''}$.

$$\begin{bmatrix} * & * & * & 0 & 0 & 0 \\ * & * & * & * & * & 0 \\ * & * & * & 0 & * & * \\ 0 & * & 0 & * & * & 0 \\ 0 & * & * & * & * & * \\ 0 & 0 & * & 0 & * & * \end{bmatrix}$$

Figure 1. Matrix $\mathbf{A} \in \mathbb{Q}^{6\times 6}$, where "$*$" denotes nonzero entries.

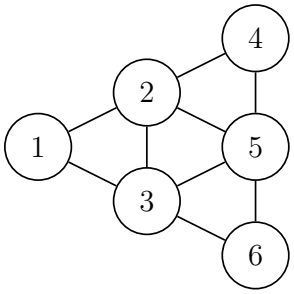

Figure 2. Graph $G = (V, E)$ constructed from the nonzero entries of $\mathbf{A}$, where $V = [6]$ and $E = \mathrm{nz}(\mathbf{A})$.

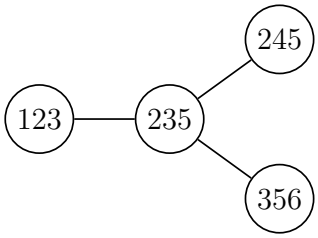

Figure 3. Tree decomposition $(T, \{X_t\}_{t\in T})$ of G. T contains four nodes, and bags are of size 3; i.e., its treewidth is 2.

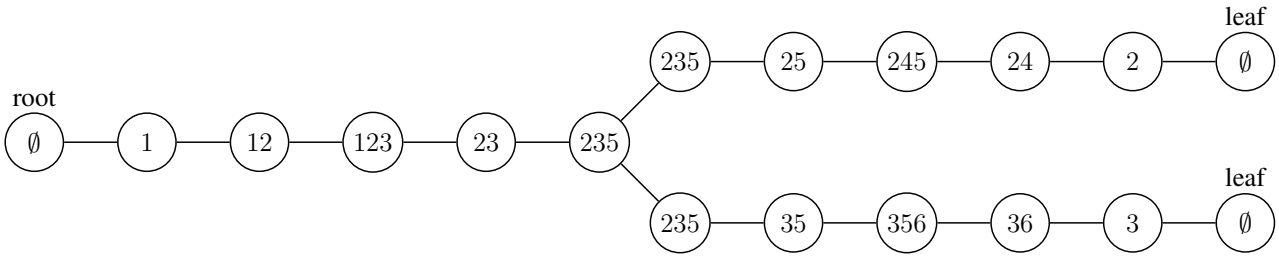

Figure 4. Nice tree decomposition of G. This decomposition is essentially identical to $(T, \{X_t\}_{t\in T})$, but this representation makes easier to develop and analyze dynamic programming algorithms.

Proof of Lemma 5.7. Since σ is a bijection, let $F_1 = \sigma^{-1}(S) \cap O_1$ $F_2 = \sigma(S) \cap O_2$, $\tau = \sigma|_{O_1 \setminus F_1}$, and $N = N(\sigma)$. Then, σ must be consistent with S and $\mathtt{st}$. The uniqueness is obvious from the definition of states. □

Proof of Lemma 5.8

The proof is separated into the following three lemmas.

Lemma E.2. *Let t be an introduce node such that $X_t = X_{t'} + v$, and $s \in [0, n]$. Given $dp_{t',s'}$ for all s', we can compute each entry of $dp_{t,s}$ in time $n^{\mathcal{O}(1)}$.*

Lemma E.3. *Let t be a forget node such that $X_t = X_{t'} - v$, and $s \in [0, n]$. Given $dp_{t',s'}$ for all s', we can compute each entry of $dp_{t,s}$ in time $w^{\mathcal{O}(w)}n^{\mathcal{O}(1)}$.*

Lemma E.4. *Let t be a join node such that $X_t = X_{t'} = X_{t''}$, and $s \in [0, n]$. Given $dp_{t',s'}, dp_{t'',s''}$ for all s', s'', we can compute each entry of $dp_{t,s}$ in time $w^{\mathcal{O}(w)}n^{\mathcal{O}(1)}$.*

Hereafter, we assume to be given a vertex ordering $\prec$ associated with a nice tree decomposition $(T, \{X_t\}_{t\in T})$ such that

1. For each introduce node t such that $X_t = X_{t'} + v$, $u \prec v$ for all $u \in V_{t'}$.
2. For each join node t such that $X_t = X_{t'} = X_{t''}$, either "$u \prec w$ for all $u \in V_{t'} \setminus X_{t'}$ and $w \in V_{t''} \setminus X_{t''}$", or "$w \prec u$ for all $u \in V_{t'} \setminus X_{t'}$ and $w \in V_{t''} \setminus X_{t''}$".

This ordering will be used to calculate the inversion number for bijections. Note that for any bijection σ, the inversion number is defined as $N(\sigma) = \{(i, j) \mid i \prec j, \sigma(i) \succ \sigma(j)\}$.

By definition, the number of bits required to express each entry of $dp_{t,s}$ is roughly bounded by $\log(2^n n!)(\text{size}(\mathbf{A}) + \text{size}(\mathbf{B})) = \mathcal{O}(n \log n(\text{size}(\mathbf{A}) + \text{size}(\mathbf{B})))$, which is a polynomial in the input size.

Proof of Lemma E.2. Let $v = X_t \setminus X_{t'}$. Remark that $V_{t'} \setminus X_{t'} = V_t \setminus X_t$, and $v \succ w$ for all $w \in V_{t'}$ By the separator property of tree decompositions, $\mathbf{A}_{\{v\},V_t \setminus X_t}$ and $\mathbf{A}_{V_t \setminus X_t,\{v\}}$ are zero-matrices. We will define a mapping $\mathtt{st} \overset{f}{\mapsto} \mathtt{st}'$ from a state for t to a state for t'. Fix a subset $S \subseteq V_t \setminus X_t$ and a state $\mathtt{st} = (O_1, O_2, F_1, F_2, \tau, N)$ for t. We first claim that if $v \in F_1$ or $v \in F_2$, then we can immediately declare that $\Upsilon_{t,A}(S, \mathtt{st}) = \Upsilon_{t,B}(S, \mathtt{st}) = 0$. This is because any bijection σ consistent with S and $\mathtt{st}$ satisfies that $\sigma(v) \in S \subseteq V_t \setminus X_t$ or $\sigma^{-1}(v) \in S \subseteq V_t \setminus X_t$. Hereafter we can safely assume that $v \notin F_1$ and $v \notin F_2$. We then handle the following five cases.

(1) if $v \notin O_1, v \notin O_2$:

Since $\mathtt{st}$ is also a state for t' in this case, we define $f(\mathtt{st}) = \mathtt{st}$. It is easy to see that

$$\Upsilon_{t,A}(S, \mathtt{st}) = \Upsilon_{t',A}(S, f(\mathtt{st})).$$

(2) if $v \in O_1 \setminus F_1, v \notin O_2$:

Let $O_1' = O_1 - v \subseteq X_{t'}$ and $O_2' = O_2 - \tau(v) \subseteq X_{t'}$. Then, we define $f(\mathtt{st}) = (O_1', O_2', F_1, F_2, \tau|_{X_{t'}}, N - \Delta N)$, where $\Delta N = N(\sigma) - N(\sigma')$ is the difference in inversion number between $\sigma \in \sigma(S, \mathtt{st})$ and $\sigma' \in \sigma(S, f(\mathtt{st}))$. Regardless of the choice of σ and σ', this ΔN is uniquely determined as $|\{w \in O_2' \mid w \succ \tau(v)\}|$. Then, for any bijection σ, $\sigma \in \sigma(S, \mathtt{st})$ if and only if $\sigma|_{V_{t'}} \in \sigma(S, f(\mathtt{st}))$. Consequently, we have that

$$\Upsilon_{t,A}(S, \mathtt{st}) = \Upsilon_{t',A}(S, f(\mathtt{st})) \cdot A_{v,\tau(v)}.$$

(3) if $v \notin O_1, v \in O_2 \setminus F_2$: By a similar argument to the case **(2)**, We can safely assume that $\tau^{-1}(v)$ exists since $\mathbf{A}_{V_{t'} \setminus X_{t'},\{v\}} = \mathbf{0}$. Let $O_1' = O_1 - \tau^{-1}(v) \subseteq X_{t'}$ and $O_2' = O_2 - v \subseteq X_{t'}$. We define $f(\mathtt{st}) = (O_1', O_2', F_1, F_2, \tau|_{X_{t'}}, N - \Delta N)$, where $\Delta N = \{w \in O_1' \mid w \succ \tau^{-1}(v)\}$ is the difference in inversion number between $\sigma \in \sigma(S, \mathtt{st})$ and $\mathtt{st}' \in (S, \mathtt{st}')$. We then have that

$$\Upsilon_{t,A}(S, \mathtt{st}) = \Upsilon_{t',A}(S, f(\mathtt{st})) \cdot A_{\tau^{-1}(v),v}.$$

(4-1) if $v \in O_1 \setminus F_1, v \in O_2 \setminus F_2, \tau(v) = v$:

Let $O_1' = O_1 - v$ and $O_2' = O_2 - v$, and we define $f(\mathtt{st}) = (O_1', O_2', F_1, F_2, \tau|_{X_{t'}}, N)$. We then have that

$$\Upsilon_{t,A}(S, \mathtt{st}) = \Upsilon_{t',A}(S, f(\mathtt{st})) \cdot A_{v,v}.$$

(4-2) if $v \in O_1 \setminus F_1, v \in O_2 \setminus F_2, \tau(v) \neq v$:

Remark that $(v, \tau(v)) \neq (\tau^{-1}(v), v)$. Let $O_1' = O_1 - v - \tau^{-1}(v)$ and $O_2' = O_2 - \tau(v) - v$, and we define $f(\mathtt{st}) = (O_1', O_2', F_1, F_2, \tau|_{X_{t'}}, N - \Delta N)$, where $\Delta N = |\{w \in O_2' \mid w \succ \tau(v)\}| + |\{w \in O_1' \mid w \succ \tau^{-1}(v)\}| + 1$ is the difference in inversion number. We then have that

$$\Upsilon_{t,A}(S, \mathtt{st}) = \Upsilon_{t',A}(S, f(\mathtt{st})) \cdot A_{v,\tau(v)} A_{\tau^{-1}(v),v}.$$

We have an analogue with regard to $\Upsilon_{t,B}$ and $\Upsilon_{t',B}$. Henceforth, for two states $\mathtt{st}_A, \mathtt{st}_B$ for t, it holds that

$$dp_{t,s} \begin{bmatrix} \mathtt{st}_A \\ \mathtt{st}_B \end{bmatrix} = dp_{t',s} \begin{bmatrix} f(\mathtt{st}_A) \\ f(\mathtt{st}_B) \end{bmatrix} A(\mathtt{st}_A) B(\mathtt{st}_B), \tag{3}$$

where $A(\mathtt{st}_A)$ is either of 1, $A_{v,\tau(v)}$, $A_{\tau^{-1}(v),v}$, $A_{v,v}$, or $A_{v,\tau(v)} A_{\tau^{-1}(v),v}$ determined based on the above case analysis. The same argument applies to $B(\mathtt{st}_B)$. Because evaluating $f(\mathtt{st}_A), f(\mathtt{st}_B), A(\mathtt{st}_A), B(\mathtt{st}_B)$ completes in $n^{\mathcal{O}(1)}$ time, so does evaluating $dp_{t,s} \begin{bmatrix} \mathtt{st}_A \\ \mathtt{st}_B \end{bmatrix}$. □

Proof of Lemma E.3. Let $v = X_{t'} \setminus X_t$. We will define a mapping $(S', \mathtt{st}') \overset{f}{\mapsto} (S, \mathtt{st})$ from a subset-state pair for t' to a subset-state pair for t. Fix a subset $S' \subseteq V_{t'} \setminus X_{t'}$ and a state $\mathtt{st}' = (O'_1, O'_2, F'_1, F'_2, \tau', N')$ for t'.

(1) if $v \notin O'_1, v \notin O'_2$: We define $S = S'$ and $\mathtt{st} = \mathtt{st}'$. Notice that $v \notin S$.

(2) if $v \in O'_1, v \in O'_2$: We first define F_1 and F_2 as follows:

- **(2-1)** $v \in F'_1, v \in F'_2$: $F_1 = F'_1 - v, F_2 = F'_2 - v$.
- **(2-2)** $v \notin F'_1, v \in F'_2$: $F_1 = F'_1, F_2 = F'_2 + \tau'(v) - v$.
- **(2-3)** $v \in F'_1, v \notin F'_2$: $F_1 = F'_1 + \tau'^{-1}(v) - v, F_2 = F'_2$.
- **(2-4)** $v \notin F'_1, v \notin F'_2, \tau(v) \neq v$: $F_1 = F'_1 + \tau'^{-1}(v), F_2 = F'_2 + \tau'(v)$.
- **(2-5)** $v \notin F'_1, v \notin F'_2, \tau(v) = v$: $F_1 = F'_1, F_2 = F'_2$.

We now define $S = S' + v$ and $\mathtt{st} = (O'_1 - v, O'_2 - v, F_1, F_2, \tau'|_{X_t}, N')$.

(3) if $v \notin O'_1, v \in O'_2$: We do not consider.

(4) if $v \in O'_1, v \notin O'_2$: We do not consider.

Our claim is the following:

Claim. *For a subset-state pair $(S, \mathtt{st})$ for t and a bijection $\sigma \in \sigma(S, \mathtt{st})$, there exists a unique subset-state pair $(S', \mathtt{st}')$ for t' such that $\sigma \in \sigma(S', \mathtt{st}')$ and $f(S', \mathtt{st}') = (S, \mathtt{st})$.*

Proof of the claim. Given $S \subseteq V_t \setminus X_t, \mathtt{st} = (O_1, O_2, F_1, F_2, \tau, N), \sigma \in \sigma(S, \mathtt{st})$, we construct a subset-state pair $(S', \mathtt{st}')$ for t' as follows:

(1) if $v \notin S$: let $S' = S$ and $\mathtt{st}' = \mathtt{st}$.

(2) if $v \in S$: We prepare F'_1 and F'_2 as follows:

- **(2-1)** $\sigma(v) \notin X_{t'}, \sigma^{-1}(v) \notin X_{t'}$: $F'_1 = F_1 + v, F'_2 = F_2 + v$.
- **(2-2)** $\sigma(v) \in X_{t'}, \sigma^{-1}(v) \notin X_{t'}$: $F'_1 = F_1, F'_2 = F_2 - \sigma(v) + v$.
- **(2-3)** $\sigma(v) \notin X_{t'}, \sigma^{-1}(v) \in X_{t'}$: $F'_1 = F_1 - \sigma^{-1}(v) + v, F'_2 = F_2$.
- **(2-4)** $\sigma(v) \in X_{t'}, \sigma^{-1}(v) \in X_{t'}, \sigma(v) \neq v$: $F'_1 = F_1 - \sigma^{-1}(v), F'_2 = F_2 - \sigma(v)$.
- **(2-5)** $\sigma(v) \in X_{t'}, \sigma^{-1}(v) \in X_{t'}, \sigma(v) = v$: $F'_1 = F_1, F'_2 = F_2$.

Let then $S' = S - v$ and $\mathtt{st}' = (O_1 + v, O_2 + v, F'_1, F'_2, \sigma|_{X_{t'}}, N)$.

It is easy to verify the requirements by case analysis; the uniqueness is obvious. □

By this claim, we have that for any $S \subseteq V_t \setminus X_t$ and state $\mathtt{st}$ for t,

$$\Upsilon_{t,A}(S, \mathtt{st}) = \sum_{\substack{S' \subseteq V_{t'} \setminus X_{t'} \\ \mathtt{st}' \text{ for } t' \\ f(S', \mathtt{st}') = (S, \mathtt{st})}} \Upsilon_{t',A}(S', \mathtt{st}').$$

We have an analogue regarding $\Upsilon_{t,B}$ and $\Upsilon_{t',B}$. Consequently, $dp_{t,s}$ can be decomposed into two sums as follows.

$$
\begin{aligned}
dp_{t,s}\begin{bmatrix}\mathtt{st}_A\\ \mathtt{st}_B\end{bmatrix} &= \sum_{S\in\binom{V_t\setminus X_t}{s}} \Upsilon_{t,A}(S,\mathtt{st}_A)\cdot\Upsilon_{t,B}(S,\mathtt{st}_B)\\
&= \sum_{S\in\binom{V_t\setminus X_t}{s}:v\notin S}\ \sum_{\substack{S'_A\subseteq V_{t'}\setminus X_{t'}\\ \mathtt{st}'_A \text{ for } t'\\ f(S'_A,\mathtt{st}'_A)=(S,\mathtt{st}_A)}}\ \sum_{\substack{S'_B\subseteq V_{t'}\setminus X_{t'}\\ \mathtt{st}'_B \text{ for } t'\\ f(S'_B,\mathtt{st}'_B)=(S,\mathtt{st}_B)}} \Upsilon_{t',A}(S'_A,\mathtt{st}'_A)\cdot\Upsilon_{t',B}(S'_B,\mathtt{st}'_B)\\
&+ \sum_{S\in\binom{V_t\setminus X_t}{s}:v\in S}\ \sum_{\substack{S'_A\subseteq V_{t'}\setminus X_{t'}\\ \mathtt{st}'_A \text{ for } t'\\ f(S'_A,\mathtt{st}'_A)=(S,\mathtt{st}_A)}}\ \sum_{\substack{S'_B\subseteq V_{t'}\setminus X_{t'}\\ \mathtt{st}'_B \text{ for } t'\\ f(S'_B,\mathtt{st}'_B)=(S,\mathtt{st}_B)}} \Upsilon_{t',A}(S'_A,\mathtt{st}'_A)\cdot\Upsilon_{t',B}(S'_B,\mathtt{st}'_B)\\
&= \sum_{S\in\binom{V_t\setminus X_t}{s}:v\notin S}\ \sum_{\substack{\mathtt{st}'_A \text{ for } t'\\ f(\mathtt{st}'_A)=\mathtt{st}_A\\ v\notin O'_{A1},v\notin O'_{A2}}}\ \sum_{\substack{\mathtt{st}'_B \text{ for } t'\\ f(\mathtt{st}'_B)=\mathtt{st}_B\\ v\notin O'_{B1},v\notin O'_{B2}}} \Upsilon_{t',A}(S,\mathtt{st}'_A)\cdot\Upsilon_{t',B}(S,\mathtt{st}'_B)\\
&+ \sum_{S\in\binom{V_t\setminus X_t}{s}:v\in S}\ \sum_{\substack{\mathtt{st}'_A \text{ for } t'\\ f(\mathtt{st}'_A)=\mathtt{st}_A\\ v\in O'_{A1},v\in O'_{A2}}}\ \sum_{\substack{\mathtt{st}'_B \text{ for } t'\\ f(\mathtt{st}'_B)=\mathtt{st}_B\\ v\in O'_{B1},v\in O'_{B2}}} \Upsilon_{t',A}(S-v,\mathtt{st}'_A)\cdot\Upsilon_{t',B}(S-v,\mathtt{st}'_B)\\
&= \sum_{\substack{\mathtt{st}'_A \text{ for } t'\\ f(\mathtt{st}'_A)=\mathtt{st}_A\\ v\notin O'_{A1},v\notin O'_{A2}}}\ \sum_{\substack{\mathtt{st}'_B \text{ for } t'\\ f(\mathtt{st}'_B)=\mathtt{st}_B\\ v\notin O'_{B1},v\notin O'_{B2}}}\ \sum_{S'\in\binom{V_{t'}\setminus X_{t'}}{s}} \Upsilon_{t',A}(S',\mathtt{st}'_A)\cdot\Upsilon_{t',B}(S',\mathtt{st}'_B)\\
&+ \sum_{\substack{\mathtt{st}'_A \text{ for } t'\\ f(\mathtt{st}'_A)=\mathtt{st}_A\\ v\in O'_{A1},v\in O'_{A2}}}\ \sum_{\substack{\mathtt{st}'_B \text{ for } t'\\ f(\mathtt{st}'_B)=\mathtt{st}_B\\ v\in O'_{B1},v\in O'_{B2}}}\ \sum_{S'\in\binom{V_{t'}\setminus X_{t'}}{s-1}} \Upsilon_{t',A}(S',\mathtt{st}'_A)\cdot\Upsilon_{t',B}(S',\mathtt{st}'_B)\\
&= \sum_{\substack{\mathtt{st}'_A \text{ for } t'\\ f(\mathtt{st}'_A)=\mathtt{st}_A\\ v\notin O'_{A1},v\notin O'_{A2}}}\ \sum_{\substack{\mathtt{st}'_B \text{ for } t'\\ f(\mathtt{st}'_B)=\mathtt{st}_B\\ v\notin O'_{B1},v\notin O'_{B2}}} dp_{t',s}\begin{bmatrix}\mathtt{st}'_A\\ \mathtt{st}'_B\end{bmatrix} + \sum_{\substack{\mathtt{st}'_A \text{ for } t'\\ f(\mathtt{st}'_A)=\mathtt{st}_A\\ v\in O'_{A1},v\in O'_{A2}}}\ \sum_{\substack{\mathtt{st}'_B \text{ for } t'\\ f(\mathtt{st}'_B)=\mathtt{st}_B\\ v\in O'_{B1},v\in O'_{B2}}} dp_{t',s-1}\begin{bmatrix}\mathtt{st}'_A\\ \mathtt{st}'_B\end{bmatrix}.
\end{aligned}
$$

Note that we let $dp_{t',s-1}[\cdot]=0$ if $s=0$. Running through all possible combinations of $\mathtt{st}'_A$ and $\mathtt{st}'_B$, we can compute $dp_{t,s}\begin{bmatrix}\mathtt{st}_A\\ \mathtt{st}_B\end{bmatrix}$ by $w^{\mathcal{O}(w)}n^{\mathcal{O}(1)}$ arithmetic operations.

□

Proof of Lemma E.4. Without loss of generality, we can assume that $v' \prec v''$ whenever $v' \in V_{t'} \setminus X_{t'}$ and $v'' \in V_{t''} \setminus X_{t''}$. We will define a mapping $(s', s'', \mathtt{st}', \mathtt{st}'') \overset{f}{\mapsto} (s, \mathtt{st})$ from two integer-state pairs for t' and t'' to an integer-state pair for t. Fix $s' \in [0, n]$, $s'' \in [0, n]$, $\mathtt{st}' = (s', O_1', O_2', F_1', F_2', \tau', N')$ for t' and $\mathtt{st}'' = (s'', O_1'', O_2'', F_1'', F_2'', \tau'', N'')$ for t''.

(1) if the following conditions are satisfied:

- $O_1' \setminus F_1' = O_1'' \setminus F_1''$.
- $O_2' \setminus F_2' = O_2'' \setminus F_2''$.
- $F_1' \cap F_1'' = \emptyset$.
- $F_2' \cap F_2'' = \emptyset$.
- $\tau' = \tau''$.

Then, for any $S' \in \binom{V_{t'} \setminus X_{t'}}{s'}, S'' \in \binom{V_{t''} \setminus X_{t''}}{s''}, \sigma' \in \sigma(S', \mathtt{st}'), \sigma'' \in \sigma(S'', \mathtt{st}'')$, let $\Delta N = N(\sigma' \cup \sigma'') - N(\sigma') - N(\sigma'')$. Regardless of the choice of $S', S'', \sigma', \sigma''$, this ΔN is uniquely determined; it holds, in fact, that

$$\Delta N = -N(\tau') + s''(|F_1'| + |F_2'|) + |F_1'| \cdot |F_2''| + |F_2'| \cdot |F_1''| + |\{(v, w) \in F_1' \times F_1'' \mid v \succ w\}| + |\{(v, w) \in F_2' \times F_2'' \mid v \succ w\}|.$$

We now define

$$\begin{aligned} s &= s' + s'', \\ \mathtt{st} &= (O_1' \cup O_1'', O_2' \cup O_2'', F_1' \uplus F_1'', F_2' \uplus F_2'', \tau'(= \tau''), N' + N'' + \Delta N). \end{aligned}$$

(2) otherwise: We do not consider.

We claim the following:

Claim. *For a subset-state pair $(S, \mathtt{st})$ for t and a bijection $\sigma \in \sigma(S, \mathtt{st})$, there exist two unique subset-state pairs $(S', \mathtt{st}')$ for t' and $(S'', \mathtt{st}'')$ for t'' such that $\sigma|_{V_{t'}} \in \sigma(S', \mathtt{st}')$, $\sigma|_{V_{t''}} \in \sigma(S'', \mathtt{st}'')$, and $f(|S'|, |S''|, \mathtt{st}', \mathtt{st}'') = (|S|, \mathtt{st})$ and $S' \uplus S'' = S$.*

Proof of the claim. Given $S \subseteq V_t \setminus X_t, \mathtt{st} = (O_1, O_2, F_1, F_2, \tau, N), \sigma \in \sigma(S, \mathtt{st})$, we construct $S', S'', \mathtt{st}', \mathtt{st}''$ as follows:

$$\begin{aligned} S' &= S \cap (V_{t'} \setminus X_{t'}) & S'' &= S \cap (V_{t''} \setminus X_{t''}) \\ O_1' &= \sigma^{-1}(S') \uplus \sigma^{-1}(O_2) & O_1'' &= \sigma^{-1}(S'') \uplus \sigma^{-1}(O_2) \\ O_2' &= \sigma(S') \uplus \sigma(O_1) & O_2'' &= \sigma(S'') \uplus \sigma(O_1) \\ F_1' &= \sigma^{-1}(S') & F_1'' &= \sigma^{-1}(S'') \\ F_2' &= \sigma(S') & F_2'' &= \sigma(S'') \\ \tau' &= \sigma|_{X_t} & \tau'' &= \sigma|_{X_t} \\ N' &= N(\sigma|_{V_{t'}}) & N'' &= N(\sigma|_{V_{t''}}) \end{aligned}$$

Observe that $V_t \setminus X_t = (V_{t'} \setminus X_{t'}) \uplus (V_{t''} \setminus X_{t''})$. Hence, it is easy to verify the requirements; the uniqueness is obvious. □

By this claim, we have that for any $S \subseteq \binom{V_t \setminus X_t}{s}$ and $\mathtt{st}$ for t,

$$\Upsilon_{t,A}(S, \mathtt{st}) = \sum_{\substack{s' \in [0,n], s'' \in [0,n] \\ \mathtt{st}' \text{ for } t', \mathtt{st}'' \text{ for } t'' \\ f(s', s'', \mathtt{st}', \mathtt{st}'') = (s, \mathtt{st})}} \sum_{\substack{S' \in \binom{V_{t'} \setminus X_{t'}}{s'} \\ S'' \in \binom{V_{t''} \setminus X_{t''}}{s''} \\ S' \uplus S'' = S}} \frac{\Upsilon_{t',A}(S', \mathtt{st}') \cdot \Upsilon_{t'',A}(S'', \mathtt{st}'')}{\mathbf{A}(\tau)}.$$

Division by $\mathbf{A}(\tau)$ is due to double count. The same relation applies to $\Upsilon_{t,B}$. Consequently, $dp_{t,s}$ can be decomposed into the sum over $dp_{t',s'}$ times $dp_{t'',s''}$ as follows.

$$
\begin{aligned}
&dp_{t,s}\begin{bmatrix}\mathtt{st}_A\\ \mathtt{st}_B\end{bmatrix}\\
&=\sum_{S\in\binom{V_t\setminus X_t}{s}}\Upsilon_{t,A}(S,\mathtt{st}_A)\cdot\Upsilon_{t,B}(S,\mathtt{st}_B)\\
&=\sum_{\substack{S\in\binom{V_t\setminus X_t}{s}\\ s'\in[0,n]\\ s''\in[0,n]}}\sum_{\substack{\mathtt{st}'_A,\mathtt{st}''_A,\mathtt{st}'_B,\mathtt{st}''_B\\ f(s',s'',\mathtt{st}'_A,\mathtt{st}''_A)=(s,\mathtt{st}_A)\\ f(s',s'',\mathtt{st}'_B,\mathtt{st}''_B)=(s,\mathtt{st}_B)}}\sum_{\substack{S'_A\in\binom{V_{t'}\setminus X_{t'}}{s'}\\ S''_A\in\binom{V_{t''}\setminus X_{t''}}{s''}\\ S'_A\uplus S''_A=S\\ S'_B\in\binom{V_{t'}\setminus X_{t'}}{s'}\\ S''_B\in\binom{V_{t''}\setminus X_{t''}}{s''}\\ S'_B\uplus S''_B=S}}\frac{\Upsilon_{t',A}(S'_A,\mathtt{st}'_A)\cdot\Upsilon_{t'',A}(S''_A,\mathtt{st}''_A)}{\mathbf{A}(\tau)}\frac{\Upsilon_{t',B}(S'_B,\mathtt{st}'_B)\cdot\Upsilon_{t'',A}(S''_B,\mathtt{st}''_B)}{\mathbf{B}(\tau)}\\
&=\sum_{s',s''}\sum_{\substack{\mathtt{st}'_A,\mathtt{st}''_A,\mathtt{st}'_B,\mathtt{st}''_B\\ f(s',s'',\mathtt{st}'_A,\mathtt{st}''_A)=(s,\mathtt{st}_A)\\ f(s',s'',\mathtt{st}'_B,\mathtt{st}''_B)=(s,\mathtt{st}_B)}}\sum_{\substack{S\in\binom{V_t\setminus X_t}{s}\\ S'\in\binom{V_{t'}\setminus X_{t'}}{s'}\\ S''\in\binom{V_{t''}\setminus X_{t''}}{s''}\\ S'\uplus S''=S}}\frac{\Upsilon_{t',A}(S',\mathtt{st}'_A)\cdot\Upsilon_{t'',A}(S'',\mathtt{st}''_A)}{\mathbf{A}(\tau)}\frac{\Upsilon_{t',B}(S',\mathtt{st}'_B)\cdot\Upsilon_{t'',B}(S'',\mathtt{st}''_B)}{\mathbf{B}(\tau)}\\
&=\sum_{\substack{s',s'',\mathtt{st}'_A,\mathtt{st}''_A,\mathtt{st}'_B,\mathtt{st}''_B\\ f(s',s'',\mathtt{st}'_A,\mathtt{st}''_A)=\mathtt{st}_A\\ f(s',s'',\mathtt{st}'_B,\mathtt{st}''_B)=\mathtt{st}_B}}\Big(\sum_{S'\in\binom{V_{t'}\setminus X_{t'}}{s'}}\frac{\Upsilon_{t',A}(S',\mathtt{st}'_A)\cdot\Upsilon_{t',B}(S',\mathtt{st}'_B)}{\mathbf{A}(\tau)}\Big)\Big(\sum_{S''\in\binom{V_{t''}\setminus X_{t''}}{s''}}\frac{\Upsilon_{t'',A}(S'',\mathtt{st}''_A)\cdot\Upsilon_{t'',B}(S'',\mathtt{st}''_B)}{\mathbf{B}(\tau)}\Big)\\
&=\frac{1}{\mathbf{A}(\tau)\cdot\mathbf{B}(\tau)}\sum_{\substack{s',s'',\mathtt{st}'_A,\mathtt{st}''_A,\mathtt{st}'_B,\mathtt{st}''_B\\ f(s',s'',\mathtt{st}'_A,\mathtt{st}''_A)=\mathtt{st}_A\\ f(s',s'',\mathtt{st}'_B,\mathtt{st}''_B)=\mathtt{st}_B}}dp_{t',s'}\begin{bmatrix}\mathtt{st}'_A\\ \mathtt{st}'_B\end{bmatrix}\cdot dp_{t'',s''}\begin{bmatrix}\mathtt{st}''_A\\ \mathtt{st}''_B\end{bmatrix}.
\end{aligned}
$$

Iterating through all possible combinations of s', s'', $\mathtt{st}'_A$, $\mathtt{st}''_A$, $\mathtt{st}'_B$ and $\mathtt{st}''_B$, we can compute each entry of dp_t by $w^{\mathcal{O}(w)}n^{\mathcal{O}(1)}$ arithmetic operations.

□

Proof of Theorem 5.10

Let $\mathbf{A}^1, \dots, \mathbf{A}^m$ be m matrices in $\mathbb{Q}^{n\times n}$, and $(T, \{X_t\}_{t\in T})$ be a nice tree decomposition for the graph $([n], \bigcup_{i\in[m]} \mathrm{nz}(\mathbf{A}^i))$, which is of width at most w and rooted at $r \in T$. We aim to compute the following quantity for each node $t \in T$:

$$\sum_{S\subseteq V_t\setminus X_t} \prod_{i\in[m]} \sum_{\substack{O_{i,1},O_{i,2}\subseteq X_t : |O_{i,1}|=|O_{i,2}| \\ \sigma_i : S\cup O_{i,1} \rightarrowtail S\cup O_{i,2}}} \mathrm{sgn}(\sigma_i)\mathbf{A}^i(\sigma_i). \tag{4}$$

In particular, this is equal to $\mathcal{Z}_m(\mathbf{A}^1, \dots, \mathbf{A}^m)$ at the root r. *States for node t* are defined as a tuple $\mathtt{st} = (O_1, O_2, F_1, F_2, \tau, N)$ in the same manner as in Section 5.2 excepting that $N \in \{0, 1\}$, which denotes the *parity* of the inversion number. We say that a bijection σ is *consistent* with $S \subseteq V_t \setminus X_t$ and $\mathtt{st} = (O_1, O_2, F_1, F_2, \tau, N)$ if **(1)** σ is a bijection $S \cup O_1 \rightarrowtail S \cup O_2$, **(2)** $F_1 = \sigma^{-1}(S) \cap O_1, F_2 = \sigma(S) \cap O_2$, **(3)** $\tau = \sigma|_{O_1\setminus F_1}$, and **(4)** $N = N(\sigma) \oplus 1$.

Lemma 5.7 can be adapted to the present definition. For each $i \in [m]$, letting $\mathtt{st}_i = (O_{i,1}, O_{i,2}, F_{i,1}, F_{i,2}, \tau_i, N_i)$ a state for node t, we can express Eq. (4) as follows:

$$\sum_{\substack{\mathtt{st}_1,\dots,\mathtt{st}_m \text{ for } t \\ s\in[0,n]}} (-1)^{N_1\oplus\cdots\oplus N_m} \sum_{S\subseteq \binom{V_t\setminus X_t}{s}} \prod_{i\in[m]} \Upsilon_{t,i}(S, \mathtt{st}_i),$$

where for all $i \in [m]$,

$$\Upsilon_{t,i}(S, \mathtt{st}_i) = \sum_{\sigma_i\in\sigma(S,\mathtt{st}_i)} \mathbf{A}^i(\sigma_i).$$

We then define the table $dp_{t,s}$ for each node $t \in T$ and $s \in [0, n]$ to store the following quantity with key $\begin{bmatrix}\mathtt{st}_1\\ \vdots \\ \mathtt{st}_m\end{bmatrix}$:

$$dp_{t,s}\begin{bmatrix}\mathtt{st}_1\\ \vdots \\ \mathtt{st}_m\end{bmatrix} = \sum_{S\in\binom{V_t\setminus X_t}{s}} \prod_{i\in[m]} \Upsilon_{t,i}(S, \mathtt{st}_i).$$

By definition $dp_{t,s}$ contains at most $w^{mw}8^{mw} = w^{\mathcal{O}(mw)}$ entries.

Assuming all arithmetic operations on the inversion number to be performed modulo 2, it is not difficult to observe that the claims in the proof of Lemmas E.2–E.4 still hold for the present definition. We thus easily extend those lemmas for the case of m matrices as follows.

Lemma E.5 (Introduce nodes)**.** *Let t be an introduce node such that $X_t = X_{t'} + v$, and $s \in [0, n]$. Given $dp_{t',s'}$ for all s', we can compute each entry of $dp_{t,s}$ in time $w^{\mathcal{O}(wm)}n^{\mathcal{O}(1)}$.*

Proof. Using a mapping f introduced in the proof of Lemma E.2, for m states $\mathtt{st}_1, \dots, \mathtt{st}_m$ for t, we have that

$$dp_{t,s}\begin{bmatrix}\mathtt{st}_1\\ \vdots \\ \mathtt{st}_m\end{bmatrix} = dp_{t',s}\begin{bmatrix}f(\mathtt{st}_1)\\ \vdots \\ f(\mathtt{st}_m)\end{bmatrix} A^1(\mathtt{st}_1)\cdots A^m(\mathtt{st}_m),$$

where $A^i(\mathtt{st}_i)$ for $i \in [m]$ is either of 1, $A^i_{v,\tau(v)}$, $A^i_{\tau^{-1}(v),v}$, $A^i_{v,v}$, or $A^i_{v,\tau(v)}A^i_{\tau^{-1}(v),v}$ determined based on the case analysis in the proof of Lemma E.2. Since evaluating $f(\mathtt{st}_i), A^i(\mathtt{st}_i)$ for all $i \in [m]$ completes in time $(mn)^{\mathcal{O}(1)}$, so does evaluating $dp_{t,s}$. □

Lemma E.6 (Forget nodes)**.** *Let t be a forget node such that $X_t = X_{t'} - v$, and $s \in [0, n]$. Given $dp_{t',s'}$ for all s', we can compute each entry of $dp_{t,s}$ in time $w^{\mathcal{O}(wm)}n^{\mathcal{O}(1)}$.*

Proof. Using a mapping f introduced in the proof of Lemma E.3, we have that for any $S \subseteq V_t \setminus X_t$, $\mathtt{st}$ for t, and $i \in [m]$,

$$\Upsilon_{t,i}(S, \mathtt{st}) = \sum_{\substack{S' \subseteq V_{t'} \setminus X_{t'} \\ \mathtt{st}'_i \text{ for } t' \\ f(S', \mathtt{st}') = (S, \mathtt{st})}} \Upsilon_{t',i}(S', \mathtt{st}').$$

Consequently, $dp_{t,s}$ can be decomposed into two sums as follows.

$$dp_{t,s}\begin{bmatrix} \mathtt{st}_1 \\ \vdots \\ \mathtt{st}_m \end{bmatrix} = \sum_{\substack{\mathtt{st}'_1 \text{ for } t' \\ f(\mathtt{st}'_1) = \mathtt{st}_1 \\ v \notin O'_{1,1}, v \notin O'_{1,2}}} \cdots \sum_{\substack{\mathtt{st}'_m \text{ for } t' \\ f(\mathtt{st}'_m) = \mathtt{st}_m \\ v \notin O'_{m,1}, v \notin O'_{m,2}}} dp_{t',s}\begin{bmatrix} \mathtt{st}'_1 \\ \vdots \\ \mathtt{st}'_m \end{bmatrix} + \sum_{\substack{\mathtt{st}'_1 \text{ for } t' \\ f(\mathtt{st}'_1) = \mathtt{st}_1 \\ v \in O'_{1,1}, v \in O'_{1,2}}} \cdots \sum_{\substack{\mathtt{st}'_m \text{ for } t' \\ f(\mathtt{st}'_m) = \mathtt{st}_m \\ v \in O'_{m,1}, v \in O'_{m,2}}} dp_{t',s}\begin{bmatrix} \mathtt{st}'_1 \\ \vdots \\ \mathtt{st}'_m \end{bmatrix}.$$

Running through all possible combinations of $\mathtt{st}'_1, \ldots, \mathtt{st}'_m$, we can compute $dp_{t,s}[\mathtt{st}_1, \ldots, \mathtt{st}_m]$ in time $w^{\mathcal{O}(wm)} n^{\mathcal{O}(1)}$. □

Lemma E.7 (Join nodes)**.** *Let t be a join node such that $X_t = X_{t'} = X_{t''}$, and $s \in [0, n]$. Given $dp_{t',s'}, dp_{t'',s''}$ for all s', s'', we can compute each entry of $dp_{t,s}$ in time $w^{\mathcal{O}(wm)} n^{\mathcal{O}(1)}$.*

Proof. Using a mapping f introduced in the proof of Lemma E.4, we have that for any $S \subseteq V_t \setminus X_t$, $\mathtt{st}$ for t, and $i \in [m]$,

$$\Upsilon_{t,i}(S, \mathtt{st}) = \sum_{\substack{s', s'' \in [0,n] \\ \mathtt{st}' \text{ for } t', \mathtt{st}'' \text{ for } t'' \\ f(s', s'', \mathtt{st}', \mathtt{st}'') = (s, \mathtt{st})}} \sum_{\substack{S' \in \binom{V_{t'} \setminus X_{t'}}{s'} \\ S'' \in \binom{V_{t''} \setminus X_{t''}}{s''} \\ S' \uplus S'' = S}} \frac{\Upsilon_{t',i}(S', \mathtt{st}') \cdot \Upsilon_{t'',i}(S'', \mathtt{st}'')}{\mathbf{A}^i(\tau)}.$$

□

Consequently, $dp_{t,s}$ can be decomposed as follows.

$$dp_{t,s}\begin{bmatrix} \mathtt{st}_1 \\ \vdots \\ \mathtt{st}_m \end{bmatrix} = \frac{1}{\mathbf{A}^1(\tau) \cdots \mathbf{A}^m(\tau)} \sum_{s', s''} \sum_{\substack{\mathtt{st}'_1, \mathtt{st}''_1 \\ f(s', s'', \mathtt{st}'_1, \mathtt{st}''_1) = \mathtt{st}_1}} \cdots \sum_{\substack{\mathtt{st}'_m, \mathtt{st}''_m \\ f(s', s'', \mathtt{st}'_m, \mathtt{st}''_m) = \mathtt{st}_m}} dp_{t',s'}\begin{bmatrix} \mathtt{st}'_1 \\ \vdots \\ \mathtt{st}'_m \end{bmatrix} \cdot dp_{t'',s''}\begin{bmatrix} \mathtt{st}''_1 \\ \vdots \\ \mathtt{st}''_m \end{bmatrix}.$$

Iterating through all possible combinations of $s', s'', \mathtt{st}'_1, \mathtt{st}''_1, \ldots, \mathtt{st}'_m, \mathtt{st}''_m$, we can compute each entry of $dp_{t,s}$ in time $w^{\mathcal{O}(wm)} n^{\mathcal{O}(1)}$.

Proof of Theorem 5.10. Our parameterized algorithm is almost identical that for Theorem 5.5. Let $(T, \{X_t\}_{t \in T})$ be a nice tree decomposition rooted at r of width at most w that has at most $\mathcal{O}(wn)$ nodes. For every leaf ℓ of T, we initialize $dp_{\ell,s}$ as

$$dp_{\ell,s}\begin{bmatrix} \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rightarrowtail\!\!\!\!\twoheadrightarrow \emptyset, 0 \\ \vdots \\ \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rightarrowtail\!\!\!\!\twoheadrightarrow \emptyset, 0 \end{bmatrix} = \begin{cases} 1 & \text{if } s = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Then, for each non-leaf node $t \in T$, we apply Lemmas E.5–E.7 to fill $dp_{t,s}$ using already-filled $dt_{t',s'}$ for all children t' of t and all $s' \in [0, n]$ in a bottom-up fashion. Completing dynamic programming, we compute $\mathcal{Z}_m$ as

$$\mathcal{Z}_m(\mathbf{A}^1, \ldots, \mathbf{A}^m) = \sum_{s, N_1, \ldots, N_m} (-1)^{N_1 \oplus \cdots \oplus N_m} \cdot dp_{r,s}\begin{bmatrix} \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rightarrowtail\!\!\!\!\twoheadrightarrow \emptyset, N_1 \\ \vdots \\ \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rightarrowtail\!\!\!\!\twoheadrightarrow \emptyset, N_m \end{bmatrix}.$$

The correctness follows from Lemmas E.5–E.7. We finally bound the complexity. Because T has at most wn nodes, each table is of size $w^{\mathcal{O}(wm)}$, and each table entry can be computed in time $w^{\mathcal{O}(wm)} n^{\mathcal{O}(1)}$ by Lemmas E.5–E.7, the whole time complexity is bounded by $w^{\mathcal{O}(wm)} n^{\mathcal{O}(1)}$. □

F. Proof in Section 5.3

Proof of Theorem 5.11. We reduce from the problem of counting all (imperfect) matchings in a bipartite graph of maximum degree 4, which is $\#\mathsf{P}$-complete (Vadhan, 2001). Let $H = (X \uplus Y, E)$ be a bipartite graph of maximum degree 4, where $E \subseteq X \times Y$ is a set of m edges between X and Y. Recall (Gillenwater, 2014)'s reduction to construct two positive semi-definite matrices $\mathbf{A}, \mathbf{B} \in \{0,1\}^{m \times m}$ indexed by edges of E so that $A_{i,j}$ is 1 if edges i, j in E share a common vertex in X and 0 otherwise, and $B_{i,j}$ is 1 if edges i, j in E share a common vertex in Y and 0 otherwise. Then, $\mathcal{Z}_2(\mathbf{A}, \mathbf{B})$ is exactly equal to the number of all (not necessarily perfect) matchings in H, and thus the $\#\mathsf{P}$-hardness follows.

By construction, $\mathbf{A}$ and $\mathbf{B}$ must be block-diagonal matrices, each block of which is a $k \times k$ all-one matrix for $k \in [4]$. Because the graph $([n], \mathrm{nz}(\mathbf{A}))$ is the disjoint union of cliques of size at most 4, whose treewidth is at most 3, it holds that $\mathrm{tw}(\mathbf{A}) \leq 3$. Similarly, $\mathrm{tw}(\mathbf{B}) \leq 3$. □

References

Affandi, R. H., Fox, E. B., Adams, R. P., and Taskar, B. Learning the parameters of determinantal point process kernels. In *ICML*, pp. 1224–1232, 2014.

Anari, N. and Gharan, S. O. A generalization of permanent inequalities and applications in counting and optimization. In *STOC*, pp. 384–396, 2017.

Anari, N., Liu, K., Gharan, S. O., and Vinzant, C. Log-concave polynomials II: High-dimensional walks and an FPRAS for counting bases of a matroid. In *STOC*, pp. 1–12, 2019.

Arnborg, S. and Proskurowski, A. Linear time algorithms for NP-hard problems restricted to partial k-trees. *Discrete Appl. Math.*, 23(1):11–24, 1989.

Arvind, V. and Kurur, P. P. Graph isomorphism is in SPP. *Inf. Comput.*, 204(5):835–852, 2006.

Bodlaender, H. L., Drange, P. G., Dregi, M. S., Fomin, F. V., Lokshtanov, D., and Pilipczuk, M. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM J. Comput.*, 45(2):317–378, 2016.

Borodin, A. and Rains, E. M. Eynard-Mehta theorem, Schur process, and their Pfaffian analogs. *J. Stat. Phys.*, 121(3–4): 291–317, 2005.

Burton, R. and Pemantle, R. Local characteristics, entropy and limit theorems for spanning trees and domino tilings via transfer-impedances. *Ann. Probab.*, 21(3):1329–1371, 1993.

Celis, L. E., Deshpande, A., Kathuria, T., Straszak, D., and Vishnoi, N. K. On the complexity of constrained determinantal point processes. In *APPROX/RANDOM*, pp. 36:1–36:22, 2017.

Celis, L. E., Keswani, V., Straszak, D., Deshpande, A., Kathuria, T., and Vishnoi, N. K. Fair and diverse DPP-based data summarization. In *ICML*, pp. 715–724, 2018.

Chen, J., Kanj, I. A., and Xia, G. Improved upper bounds for vertex cover. *Theor. Comput. Sci.*, 411(40–42):3736–3756, 2010.

Cygan, M., Fomin, F. V., Kowalik, Ł., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., and Saurabh, S. *Parameterized Algorithms*. Springer, 2015.

Djolonga, J. and Krause, A. From MAP to marginals: Variational inference in Bayesian submodular models. In *NIPS*, pp. 244–252, 2014.

Downey, R. G. and Fellows, M. R. *Parameterized Complexity*. Springer, 2012.

Edmonds, J. Systems of distinct representatives and linear algebra. *J. Res. Natl. Bur. Stand.*, 71B:241–245, 1967.

Eiben, E., Ganian, R., Kanj, I., and Szeider, S. The parameterized complexity of cascading portfolio scheduling. In *NeurIPS*, pp. 7666–7676, 2019.

Ganian, R., Kanj, I., Ordyniak, S., and Szeider, S. Parameterized algorithms for the matrix completion problem. In *ICML*, pp. 1656–1665, 2018.

Garey, M. R. and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

Gartrell, M., Paquet, U., and Koenigstein, N. Low-rank factorization of determinantal point processes. In *AAAI*, pp. 1912–1918, 2017.

Gartrell, M., Brunel, V.-E., Dohmatob, E., and Krichene, S. Learning nonsymmetric determinantal point processes. In *NeurIPS*, pp. 6715–6725, 2019.

Gillenwater, J. A. *Approximate Inference for Determinantal Point Processes*. PhD thesis, University of Pennsylvania, 2014.

Gong, B., Chao, W., Grauman, K., and Sha, F. Diverse sequential subset selection for supervised video summarization. In *NIPS*, pp. 2069–2077, 2014.

Gotovos, A., Hassani, S. H., and Krause, A. Sampling from probabilistic submodular models. In *NIPS*, pp. 1945–1953, 2015.

Gurvits, L. On the complexity of mixed discriminants and related problems. In *MFCS*, pp. 447–458, 2005.

Halin, R. S-functions for graphs. *J. Geom.*, 8(1-2):171–186, 1976.

Harvey, D., van der Hoeven, J., and Lecerf, G. Even faster integer multiplication. *J. Complexity*, 36:1–30, 2016.

Jerrum, M., Sinclair, A., and Vigoda, E. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51(4):671–697, 2004.

Kogan, G. Computing permanents over fields of characteristic 3: Where and why it becomes difficult (extended abstract). In *FOCS*, pp. 108–114, 1996.

Krause, A., Singh, A., and Guestrin, C. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, 9:235–284, 2008.

Kulesza, A. and Taskar, B. k-DPPs: Fixed-size determinantal point processes. In *ICML*, pp. 1193–1200, 2011.

Kulesza, A. and Taskar, B. Determinantal point processes for machine learning. *Found. Trends Mach. Learn.*, 5(2–3): 123–286, 2012.

Li, C., Jegelka, S., and Sra, S. Fast DPP sampling for Nyström with application to kernel methods. In *ICML*, pp. 2061–2070, 2016.

Macchi, O. The coincidence approach to stochastic point processes. *Adv. Appl. Probab.*, 7(1):83–122, 1975.

Mariet, Z. E., Sra, S., and Jegelka, S. Exponentiated strongly Rayleigh distributions. In *NeurIPS*, pp. 4464–4474, 2018.

O'Donnell, R. and Ta, F. Semidefinite programs and the max-cut problem, lecture 10. `http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15859-f11/www/notes/lecture10.pdf`, 2011.

Robertson, N. and Seymour, P. D. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986.

Robinson, J., Sra, S., and Jegelka, S. Flexible modeling of diversity with strongly log-concave distributions. In *NeurIPS*, pp. 15199–15209, 2019.

Schrijver, A. *Theory of Linear and Integer Programming*. Wiley–Interscience Series in Discrete Mathematics and Optimization. Wiley, 1999.

Vadhan, S. P. The complexity of counting in sparse, regular, and planar graphs. *SIAM J. Comput.*, 31(2):398–427, 2001.

Valiant, L. G. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8(2):189–201, 1979.

van der Maaten, L. and Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.*, 9(11):2579–2605, 2008.