Planar Perfect Matching Counting is as Hard as Determinants

Radu Curticapean[*] Jiaheng Wang[†]

Abstract

In the 1960s, Fisher, Kasteleyn and Temperley designed an ingenious algorithm for computing the partition function of the *dimer model*, or equivalently, for counting perfect matchings in edge-weighted planar graphs (*Philos. Mag.* 1961; *J. Mathematical Phys.* 1963). This *FKT algorithm* later became the foundation for Valiant's *holographic algorithms* (FOCS 2004; *SIAM J. Comput.* 2008), which motivated the study of counting problems under the Holant framework. Combined with an algorithm by Yuster (FOCS 2008), the FKT algorithm allows us to count edge-weighted perfect matchings in planar n-vertex graphs with $\tilde{O}(n^{\omega/2})$ arithmetic operations, where $\omega < 2.372$ is the matrix multiplication exponent.

We prove a corresponding lower bound: Over algebraic circuits and other sufficiently strong computational models, perfect matchings in edge-weighted n-vertex planar graphs G cannot be counted in $O(n^{\omega/2-\epsilon})$ arithmetic operations. This confirms the optimality of Yuster's algorithm. Our bound holds even when G is an edge-weighted square grid.

1 Introduction

The complexity class #**P** introduced by Valiant [Val79b] shows a fundamental separation between decision and counting problems, most prominently for the *perfect matching* problem: while the existence of perfect matchings in graphs can be determined in polynomial time [Edm65], counting them is #**P**-complete [Val79b]. Given this intractability, the Fisher–Kasteleyn–Temperley (FKT) algorithm [Kas63, TF61] stands as a remarkable anomaly, as it allows us to count perfect matchings in *planar* graphs G in polynomial time. The FKT algorithm achieves this by introducing carefully chosen signs into the adjacency matrix of G and using an ingenious cancellation property to ensure that the determinant of the resulting matrix counts (pairs of) perfect matchings in G. The FKT method also works when the input graph G is edge-weighted and matchings are weighted by the product of involved edge-weights; we denote this computational problem as #PlanarPM.

Beyond its implications in statistical physics such as *Exactly Solved Models* [Bax82, Wel93], the FKT algorithm is the foundation of Valiant's *holographic algorithms*, which are polynomial-time algorithms for problems that appear to evade classical techniques in algorithm design [Val06, Val08, CFGW22, Bac21]. These algorithms gave birth to the ongoing project of classifying the complexity of counting problems under the *Holant* framework, which led to a plethora of research papers, monographs and a textbook over the last two decades, e.g., [CL11, HL16, SC20, CLX20, CFGW22, CC17]. In most of these results, the planar cases are solved by an algorithmic template that ultimately involves the FKT algorithm. Indeed, every tractable planar *#CSP* can be solved along these lines [CF22]; this was discovered while studying to which extent the FKT algorithm is *universal* for counting problems on planar graphs. (It is not universal for *Holant problems* [CFGW22].) In this paper, we study a different question—

[*]University of Regensburg and IT University of Copenhagen. Funded by the European Union (ERC, CountHom, 101077083). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency.

[†]University of Regensburg and University of Helsinki. Supported by the ERC project CountHom and the Helsinki Institute of Information Technology (HIIT).

Is the FKT algorithm optimal?

Given the fundamental role of the problem #PlanarPM in statistical physics and theoretical computer science, e.g., in the form of Holant problems, it is important to determine its precise complexity. The FKT algorithm readily gives an upper bound: On an n-vertex planar graph, the running time of this algorithm is dominated by the evaluation of a particular $n \times n$ determinant $\det(A)$. There are several options for computing $\det(A)$:

- Algorithms for generic determinants allow us to evaluate $\det(A)$ with $\tilde{O}(n^{\omega})$ arithmetic operations, where $\omega < 2.372$ is the *matrix multiplication exponent.*[1]
- However, the matrix A inherits a great deal of structure from the planarity of the input graph G. This can be exploited, e.g., using the *nested dissection method* [Geo73, LRT79] and additional insights [Wil97, MS06, YZ07, Yus08], to evaluate $\det(A)$ in $\tilde{O}(n^{\omega/2})$ arithmetic operations.

The exponent $\omega_{\mathrm{MM}}/2$ is a natural barrier for the nested dissection method, since it relies on planar separators, which may require $O(\sqrt{n})$ vertices in the worst case. Consequently, any attempt at further improving the running time would likely need to bypass the separator-based approach.

Note however that this algorithmic strategy might already be optimal, and we merely do not know yet: Our current knowledge permits the possibility of $\omega_{\mathrm{MM}} = 2$, which would imply an optimal exponent of $\omega_{\mathrm{MM}}/2 = 1$. Conversely, an $\Omega(n^{1+\epsilon})$ lower bound on #PlanarPM would directly translate into $\omega_{\mathrm{MM}} > 2$. Such a lower bound however appears to be out of reach, even conditioned on the strong exponential-time hypothesis [IP01], the APSP conjecture [Wil18], or other popular fine-grained conjectures [VW18].

Our results

Although the current fine-grained complexity landscape offers little explanation for the $\omega/2$ barrier for the problem #PlanarPM, we can still obtain a meaningful statement about its complexity by relating it to *generic* determinants: By showing that generic $n \times n$ determinants can be reduced to #PlanarPM on grid graphs of side-length $O(n)$, we prove the following lower bound:

Theorem 1. *No algorithm solves #PlanarPM on planar n-vertex edge-weighted graphs with $O(n^{\omega/2-\epsilon})$ arithmetic operations for any $\epsilon > 0$, even when the input graph is an edge-weighted grid. Here, $\omega = \omega_{\mathrm{DET}}$ is the matrix determinant constant.*

Consequently, over $\mathbb{R}$, the $\tilde{O}(n^{\omega_{\mathrm{MM}}/2})$ algorithms by Yuster and Zwick [YZ07, Yus08] are asymptotically optimal (up to polylogarithmic factors) in computational models with $\omega_{\mathrm{DET}} = \omega_{\mathrm{MM}}$. This includes standard algebraic models that support the Baur–Strassen Theorem [BS83], e.g., algebraic circuits and algebraic branching programs.

Regarding the role of edge-weights, we note that the graphs constructed in the proof of Theorem 1 contain edges of negative weight. It is possible to remove such edges, but the running time overhead caused in the reduction would lead to a weaker bound. Over prime fields $\mathbb{F}_p$, a modified stretching/thickening argument [JVW90] can be implemented with only a constant-factor overhead, yielding the following theorem:

Theorem 2. *For any fixed prime $p \geq 2$ and $\epsilon > 0$, no algorithm solves #PlanarPM over $\mathbb{F}_p$ on unweighted planar n-vertex graphs with $O(n^{\omega/2-\epsilon})$ arithmetic operations. Here, $\omega = \omega_{\mathrm{DET}}$ is the matrix determinant constant.*

[1] To be more precise, but a bit technical: We can further replace the matrix multiplication constant with the *matrix determinant constant* ω_{DET} since the algorithm only asks for the determinant. The relation between ω_{DET} and ω_{MM} is discussed later at the end of the introduction.

Connections to Algebraic Complexity Theory

Determinants are complete for the algebraic complexity class **VBP**, which captures computations by polynomial-sized *algebraic branching programs* or *skew* arithmetic circuits.[2] Dropping skewness yields the class **VP**, the algebraic version of $\mathbf{P}/_{\text{poly}}$ [Val79a]. It is conjectured that **VBP** ≠ **VP**.

It is known that determinants capture skew circuits and **VBP** with low overhead. In the following, we write $\det_m$ for the determinant of an $m \times m$ matrix with generic indeterminates x_{ij}; this is a multivariate polynomial of degree m. The following is known:

- The determinant $\det_m$ admits $O(m^4)$-size skew circuits, e.g., by Berkowitz's method [Ber84, Sam42] or the beautiful combinatorial approach of Mahajan and Vinay [MV97].
- Every function with a skew circuit of size m is a projection of $\det_{m+1}$ [Tod92, MP08].

Since the FKT algorithm establishes a reduction from #PlanarPM to determinants, one may wonder whether the converse reduction is also possible. This seems plausible *a priori*, because the determinant $\det(A)$ arising in the FKT algorithm is, up to signs, the adjacency matrix of an arbitrary edge-weighted planar graph G. Indeed, Flarup, Koiran and Lyaudet [FKL07] already established 20 years ago that every polynomial p with size-n skew circuits is a projection of #PlanarPM with $O(n^2)$-vertex planar graphs. In this projection, edge-weights are univariate linear functions in the variables of p. Composing this result with the **VBP**-completeness reduction of the determinant, one can express $\det_m$ as #PlanarPM on a planar graph on $O(m^8)$ vertices. Thus, up to polynomial factors, the determinant and #PlanarPM express the same polynomials through projections.

To study the *fine-grained complexity* of #PlanarPM however, the $O(m^8)$ blowup is prohibitive. Towards our main theorem, we establish a reduction with the *optimal* blowup of $O(m^2)$. This bound is indeed optimal for the trivial reason that n-vertex planar graphs have $O(n)$ edges and therefore require $\Omega(m^2)$ vertices to capture the m^2 variables in $\det_m$ by distinct edges.

Theorem 3 (Optimal Expressiveness). *Let $m \in \mathbb{N}$ and let $\boldsymbol{X} = (x_{ij})$ be an $m \times m$ matrix with indeterminates. Then there is an edge-weighted $3m \times 5m$ grid graph $G_{\boldsymbol{X}}$ with edge weights $\{-1, 0, 1\} \cup \{1 - x_{ij} : i, j \in [m]\}$ such that, as polynomials,*

$$\det(\boldsymbol{X}) \equiv \mathrm{pm}(G_{\boldsymbol{X}}).$$

We remark that the negative edge-weights introduced in Theorem 3 are necessary, as the determinant can be negative even on 0-1 matrices, while the plain *unweighted* count of perfect matchings in a graph is of course always nonnegative.

Proof Overview

Our proof is fully elementary and self-contained. In the proof, we use a minimal subset of the vast theory of *Holant problems*, which we outline below. In machine learning and physics, such problems are also known as (contractions of) *tensor networks* [CLS⁺24, HPM⁺19, EV15].

Holant problems. Holant problems count weighted Boolean assignments to the edges of a graph. The weight of an assignment is determined by local factors contributed by the vertices. More formally, consider a graph $G = (V, E)$ equipped with a set of functions $\mathcal{F} = \{f_v\}$, so-called *signatures*, such that $f_v : \{0,1\}^{\deg(v)} \to \mathbb{C}$ on vertex v obtains as input an assignment to the edges incident with v. Any 0-1

[2] A skew circuit involves at least one input gate at each multiplication gate. The related *weakly-skew* circuits and algebraic branching programs turn out to have *exactly* the same computational power [KK08, Section 2]; see also [Bür24, Remark 2.22].

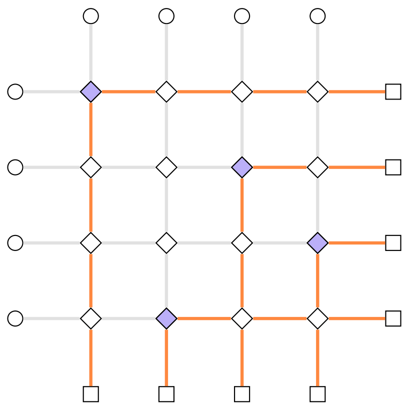

Figure 1: A blue diamond indicates a marshaller. Orange lines indicate light beams. The circles and squares indicate sentinel vertices that are useful in simplifying the Holant construction. A constellation is valid iff each diamond is in state , , , , or , no circle has an incident beam, and each square has an incident beam.

edge assignment σ gives rise to a weight, which is the product of all f_v under σ. The *Holant* on $(G;\mathcal{F})$ is the sum of these weights over all possible σ:

$$\mathrm{holant}(G;\mathcal{F}) = \sum_{\sigma:E\to\{0,1\}} \prod_{v\in V} f_v(\sigma|_{E(v)}).$$

As an example, the problem of counting perfect matchings in a graph G can be viewed as a Holant problem: Associate to each vertex $v \in V(G)$ the signature $f_v : \{0,1\}^{\deg(v)} \to \{0,1\}$ with

$$f_v(a) = 1 \quad \text{if and only if the Hamming weight of } a \text{ is } 1.$$

Counting perfect matchings with weights can also be expressed as a Holant problem: It suffices to subdivide edges once and place signatures on the subdivision vertices that capture their edge-weight.[3] When G is bipartite, counting weighted perfect matchings in G amounts to evaluating the *permanent* of the bi-adjacency matrix of G.

Permanents as Holants on grids. The above transformation from permanents into Holant problems directly mirrors the structure of the input graph into the Holant problem instance. Recent work on the complexity of multilinear forms [BCK+26] established a more input-agnostic transformation: As shown in [BCK+26], the $n \times n$ permanent can be expressed as the Holant of an $n \times n$ *grid graph* G. In this construction, the vertices of the grid G correspond directly to the entries of A, and appropriate signatures ensure that the Holant counts combinatorial structures that correspond directly to row-column permutations $\pi \in S_n$, each weighted by $\prod_{i=1}^{n} a_{i,\pi i}$.

The structures counted in the Holant problem have a clean combinatorial interpretation, see also Figure 1: They are constellations of n "marshallers" and n vertical and n horizontal "light beams" on the grid, subject to the following constraints:[4]

1. Each marshaller sends a horizontal beam to the right and a vertical beam downwards. It contributes the factor $a_{i,j}$ to the weight when located at position (i,j).

[3] On the subdivision vertex s induced by an edge of weight w, place $f(00) = 1$ and $f(01) = f(10) = 0$ and $f(11) = w$.

[4] This construction is very similar to a *non-attacking rook placement*. The marshallers could be viewed as the main characters in the construction, but the beams are crucial to ensure a clean formulation as a Holant problem.

2. The beams of any marshaller may not hit another marshaller.

3. Beams may cross without any consequences.

It is easy to see that the valid constellations of n marshallers on the $n \times n$ grid correspond bijectively to permutations. Indeed, every such constellation represents a permutation matrix P_π overlaid on A. The weight of a constellation is the product of weights from marshallers, and thus equal to $\prod_{i=1}^{n} a_{i,\pi i}$.

The relevant constraints on marshallers and beams can be enforced straightforwardly with appropriate Holant signatures: If an edge is assigned 1, it is understood as carrying a beam.

Determinants as Holants on grids. In this paper, we observe that the constellations described above also capture the *determinant*: Indeed, in the constellation corresponding to a permutation π, beam crossings occur *precisely* at the inversions of π. Thus, to express the determinant as a Holant problem over a grid graph, one only needs to introduce a negative sign for crossing beams. In other words, beam crossings are no longer consequence-free; they are still allowed, but they must be accounted for by a negative sign. On the level of Holants, this is easily reflected by modifying the signature output for the state  from $+1$ to -1.

The modified signature f obtained this way satisfies the *Matchgate Identities* [Val02b, Val02a, CC07, CCL09, CG14], which implies the existence of a planar graph H with $O(1)$ vertices and four dangling edges that simulates f: For every subset S of dangling edges, the signature value $f(S)$ equals the weighted number of perfect matchings in the graph $H - S$ obtained from H by deleting S and all incident vertices. Cai and Gorenstein [CG14] give an elegant explicit construction of H, which was also used, e.g., in [CX22]. The final planar graph $G_{\boldsymbol{X}}$ to simulate the determinant is then obtained by replacing every vertex of signature f in the grid by a copy of the gadget H. The resulting graph again fits neatly into a grid.

2 Preliminaries

2.1 Perfect Matchings

A perfect matching of a graph $G = (V, E)$ is an edge subset $M \subseteq E$ that every vertex $v \in V$ appears exactly once in M. On input an edge-weighted graph $G = (V, E, w)$, the computational task #PlanarPM asks for a weighted count of perfect matchings in G, where each matching is weighted by the product of its edge weights. Formally:

Name: #PlanarPM

Instance: A simple undirected planar graph $G = (V, E, w)$ with edge weights $w : E \to \mathbb{C}$.

Output: Weighted sum of all perfect matchings $M \subseteq E$:

$$\mathrm{pm}(G) \coloneqq \sum_{M} \prod_{e \in M} w(e).$$

2.2 Determinant versus Matrix Multiplication

The matrix multiplication constant ω_{MM} is defined as the smallest real number such that n-by-n matrix multiplication admits an algorithm using $n^{\omega_{\mathrm{MM}}+o(1)}$ arithmetic operations. The matrix determinant constant ω_{DET} is defined analogously. A classical result states that $\omega_{\mathrm{MM}} = \omega_{\mathrm{DET}}$ under standard algebraic models, and both directions are highly non-trivial:

1. Bunch and Hopcroft showed $\omega_{\mathrm{MM}} \geq \omega_{\mathrm{DET}}$ by means of a carefully-designed LU decomposition and analysis [Str69, BH74].

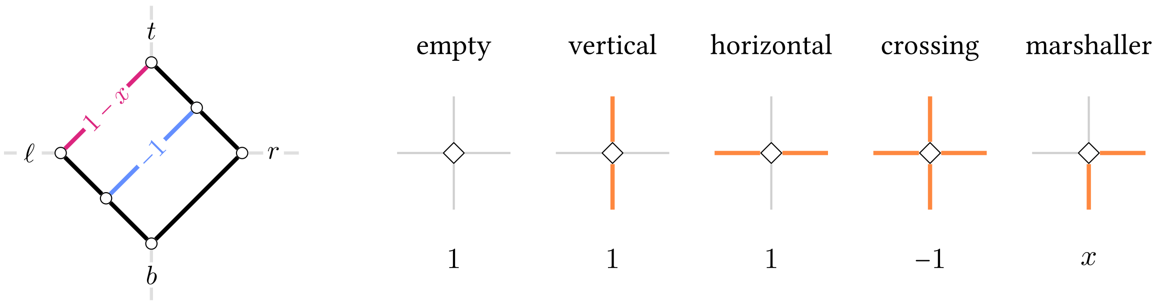

Figure 2: The planar H-gadget.

2. Baur and Strassen established $\omega_{\mathrm{MM}} \leq \omega_{\mathrm{DET}}$ by computing the partial derivatives together with the original function, and then applying Cramer's Rule [BS83].

However, the relation between matrix multiplication and matrix determinant is not known beyond algebraic models, e.g., in terms of bit complexity. It is worth remarking that Yuster's algorithm runs in $O(n^{\omega_{\mathrm{MM}}/2+1})$ bit operations.

Remark. There is no need to specify the underlying field in Theorem 3, because the matrix entries are treated as indeterminates. Theorem 1 therefore works for every field; one just needs to specify the field for defining the matrix determinant constant.

On the other hand, it is a major open problem whether these constants are field-independent, i.e., whether $\omega(F)$ agrees for all fields F. All existing techniques for bounding ω are field-independent, and it is known that $\omega_{\mathrm{MM}}(F) = \omega_{\mathrm{MM}}(F')$ if $\mathrm{char}(F) = \mathrm{char}(F')$ [Sch81, Theorem 2.8], which also suggests field independency.

3 The reduction

We first describe the construction of individual gadgets and then show how to compose the gadgets to obtain the overall reduction.

3.1 Gadgets

Our reduction is based around the planar gadget H shown in Figure 2, which has 6 vertices, 7 internal edges, and 4 external *dangling edges*, which we denote by t, r, b, ℓ, going clockwise from the top. If a dangling edge is assigned 0, the attaching vertex has to be *matched within* the gadget, otherwise it must be left *unmatched within* the gadget. Assignments are represented as 4-bit strings in the order t, r, b, ℓ, e.g., the string 0110 represents the assignment $t = \ell = 0, r = b = 1$. The weight of H on a given assignment to the dangling edges is the weighted count of perfect matchings in $H - S$, where S is the set of endpoints of all dangling edges that are assigned 1.

Definition 4 (states of a gadget, see Figure 2)**.**

 $:=$ 0000 (empty)

 $:=$ 1010 (vertical)

 $:=$ 0101 (horizontal)

 $:=$ 1111 (crossing)

 $:=$ 0110 (marshaller)

The 11 of 16 remaining states from $\{0, 1\}^4$ are considered *invalid*.

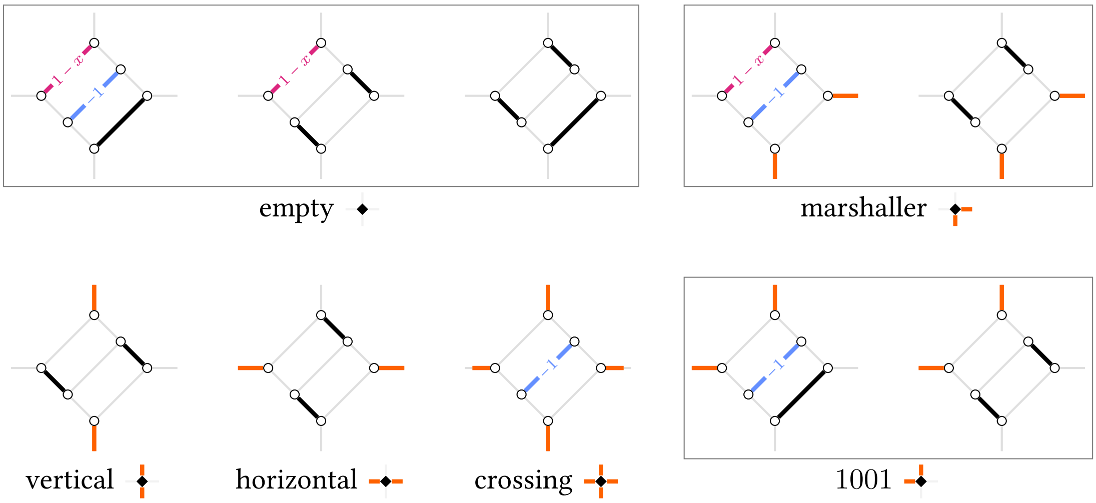

Figure 3: Computing the weight of the H-gadget in different states. Cases not shown here have no internal perfect matching.

Lemma 5. *The weight of an H-gadget is*

$$\begin{cases} 1, & \text{if } H \text{ is } \ , \ \text{ or } \ , \\ -1, & \text{if } H \text{ is } \ , \\ x, & \text{if } H \text{ is } \ , \text{ and} \\ 0, & \text{if } H \text{ is invalid.} \end{cases}$$

Proof. For assignments a of odd Hamming weight, the graph $H - S$ has an odd number of vertices and thus no perfect matching, so the weight of H under a is zero. This leaves 8 cases to consider. Among these, the cases 1100 and 0011 also admit no perfect matching in $H - S$. For the remaining 6 cases, Figure 3 enumerates all the possible matchings and their weights. Notably, in the invalid state 1001 , the two perfect matchings have weights -1 and 1, and they hence cancel. □

To simplify the construction, we introduce two "sentinel" gadgets to be placed at the grid border. They are not necessary, but they allow us to state the proof in a uniform way.

- The *single-edge* gadget contains vertices u, v connected by an edge, with a dangling edge at v.
- The *single-vertex* gadget contains a single vertex v with a dangling edge.

Both gadgets can be easily removed from the construction without changing the number of perfect matchings.

3.2 Graph construction

Next, we construct an edge-weighted graph $G_{\boldsymbol{X}}$ by the following recipe. See Figure 1 for the outcome when $\boldsymbol{X}$ is a 4-by-4 matrix.

1. For all $i, j \in [m]$, introduce a fresh H-copy $H_{i,j}$. Replace the indeterminate in $H_{i,j}$ by x_{ij}.
2. For all $i \in [m]$ and $j \in [m-1]$, join the r-edge of $H_{i,j}$ with the ℓ-edge of $H_{i,j+1}$.

Figure 4: The gadget $H_{j,\pi i}$ is in state iff (i, j) forms an inversion, i.e., $i < j$ and $\pi i > \pi j$. The left part shows an inversion, the right part shows a non-inversion.

3. For all $i \in [m - 1]$ and $j \in [m]$, join the b-edge of $H_{i,j}$ with the t-edge of $H_{i+1,j}$.
4. For all $i \in [m]$, introduce a fresh single-edge gadget, and join its dangling edge with the ℓ-edge of $H_{i,1}$. Perform the same for t-edge of $H_{1,j}$, for all $j \in [m]$.
5. For all $i \in [m]$, introduce a fresh single-vertex gadget, and join the dangling edge with the r-edge of $H_{i,m}$. Perform the same for the b-edge of $H_{m,j}$, for all $j \in [m]$.

Consequently, each row and column contains $m + 1$ edges, all unweighted, that are formed by joining two dangling edges. Let τ be a 0-1 assignment to all of these edges; it determines the states of all H-gadgets. Under this assignment, the weighted perfect matching count $\mathrm{wt}(\tau) \coloneqq \mathrm{pm}(G_{\boldsymbol{X}} \mid \tau)$ is the product of the weights of all gadgets under their respective local sub-assignments of τ.

3.3 Analysis of the construction

In the following, denote the set of all non-vanishing assignments τ by

$$\Gamma \coloneqq \{\tau : \mathrm{wt}(\tau) \not\equiv 0\}.$$

These non-vanishing assignments are characterised by the following series of propositions.

Proposition 6. *Let $\tau \in \Gamma$ be non-vanishing and $i \in [m]$. For any $j \in [m]$, let s_j be the assignment of τ to the r-edge of $H_{i,j}$, and let s_0 be the assignment of τ to the ℓ-edge of $H_{i,1}$. Then there exists j^* in $[m]$ such that $s_j = 0$ for all $j < j^*$ and $s_j = 1$ for all $j \geq j^*$.*

Proof. No gadget is invalid since $\tau \in \Gamma$. This means there is no such j that $s_j = 1$ while $s_{j+1} = 0$. Further, the single-edge gadget on the i-th row ensures that $s_0 = 0$, and the single-vertex gadget on the i-th row ensures that $s_m = 1$. This leaves the patterns in the proposition the only possibilities. □

A column version of Proposition 6 holds analogously. Therefore, each row and column has exactly one gadget. Furthermore, once the positions of all these gadgets are fixed, the only non-vanishing mapping τ is also determined. This yields the following bijection.

Corollary 7. *There is a bijection $h : \Gamma \to S_m$. If $h(\tau) = \pi$, then $H_{i,\pi i}$ is the gadget of τ on the i-th row, for any $i \in [m]$.*

It is left for us to compute $\mathrm{wt}(\tau)$. In the following, recall that an ordered pair i, j forms an inversion if $i < j$ and $\pi i > \pi j$.

Lemma 8. *Let $\tau \in \Gamma$ and $\pi = h(\tau)$. The number of inversions of π equals the number of gadgets in τ.*

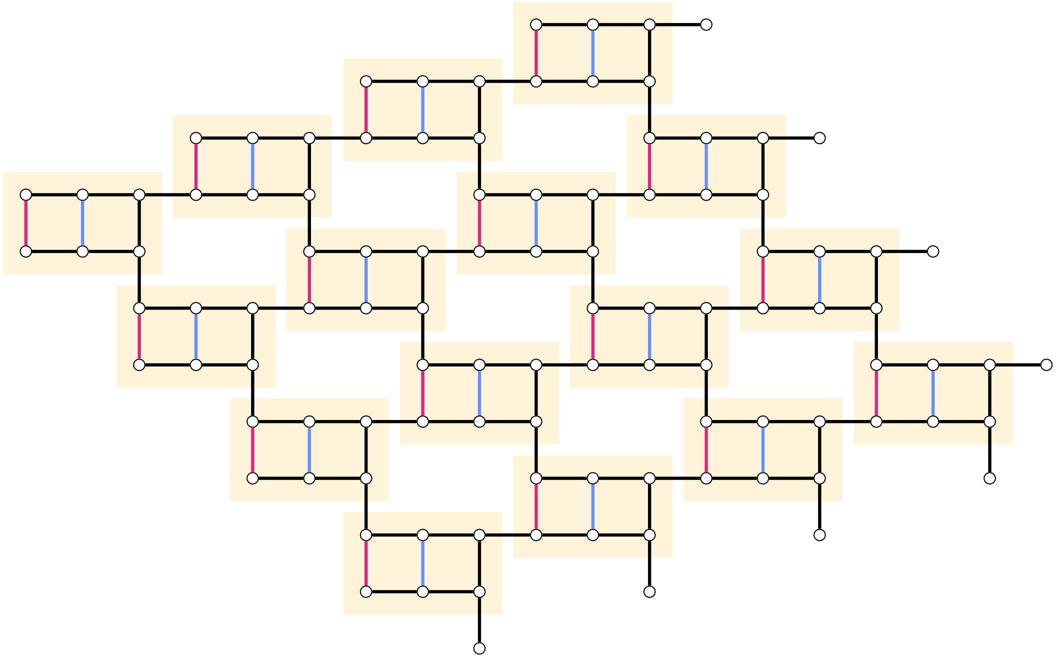

Figure 5: A grid embedding of the graph $G_{\boldsymbol{X}}$, where $\boldsymbol{X}$ is a 4-by-4 matrix.

Proof. The mapping $\varphi : (i,j) \mapsto (j, \pi i)$ clearly is a bijection $[m]^2 \to [m]^2$. We show that φ is a bijection between inversions in π and gadgets in τ. (See also Figure 4 for a quick illustration.)

- Suppose i, j forms an inversion. Then $i < j$ and the column indices of the gadgets on row i and j satisfy $\pi i > \pi j$, leaving $H_{j,\pi i}$ in state .
- Conversely, suppose $H_{j,\pi i}$ is a gadget. The gadget on row j lies at position $(j, \pi j)$, which implies $\pi i > \pi j$. The gadget on column πi is at $(\pi^{-1}\pi i, \pi i) = (i, \pi i)$, and therefore $i < j$. Hence (i,j) forms an inversion. □

3.4 Proof of main theorems

All parts are now ready to prove the main theorems.

Proof of Theorem 3. For any $\tau \in \Gamma$, using Lemmas 5 and 8 to compute the weight, we have

$$\mathrm{wt}(\tau) \equiv (-1)^{\mathrm{inv}(\pi)} \prod_{i=1}^{m} x_{i,\pi i}$$

with the bijection $\pi = h(\tau)$ from Corollary 7. Since only assignments $\tau \in \Gamma$ contribute to $\mathrm{pm}(G_{\boldsymbol{X}})$,

$$\mathrm{pm}(G_{\boldsymbol{X}}) \equiv \sum_{\tau \in \Gamma} \mathrm{wt}(\tau) \equiv \sum_{\pi \in S_m} (-1)^{\mathrm{inv}(\pi)} \prod_{i=1}^{m} x_{i,\pi i} \equiv \det(\boldsymbol{X}).$$

Finally, the graph $G_{\boldsymbol{X}}$ is indeed the subgraph of an $3m \times 5m$ grid. By assigning weight 0 to all unused grid edges, $G_{\boldsymbol{X}}$ may also be viewed as an edge-weighted $3m \times 5m$ grid graph. Figure 5 depicts an exemplary grid embedding of $G_{\boldsymbol{X}}$ up to single-edge gadgets. □

Figure 6: Replacing an edge uv of weight $w \neq 1$ with gadget M_k for $k = (w-1)^{-1}$ over $\mathbb{F}_p$.

Proof of Theorem 1. Assume $\omega = \omega_{\mathrm{DET}} > 2$, as the statement holds vacuously otherwise. Suppose there is an algorithm $\mathcal{A}$ that solves #PLANARPM in time $O(n^{\omega/2-\epsilon})$ for $\epsilon > 0$. Given $A \in F^{m\times m}$, construct the graph G_A obtained from Theorem 3 by substituting $x_{ij} = a_{ij}$. Then $\mathrm{pm}(G_A) = \det(A)$, and running $\mathcal{A}$ on G_A outputs $\det(A)$ in time $O(m^{\omega-2\epsilon})$, contradicting the definition of ω. The graph G_A is a grid subgraph; see Figure 5. ☐

To show Theorem 2, we replace the edges of weights $1 - x_{ij}$ or -1 with the gadget M_k shown in Figure 6. The following proposition is immediate.

Proposition 9. *The graph M_k has $k+1$ perfect matchings when both dangling edges are assigned* 0*, and k perfect matchings when both dangling edges are assigned* 1.

Proof of Theorem 2. The theorem is trivial over $\mathbb{F}_2$, so assume $p \geq 3$. All equations are taken over $\mathbb{F}_p$.

First, replace all indeterminates in $G_{\boldsymbol{X}}$ with the concrete values from the input matrix. Then, for any edge $e = (u, v)$ with weight $w(e) \notin \{0, 1\}$ in $G_{\boldsymbol{X}}$, introduce a fresh gadget M_k with $k = (w(e)-1)^{-1} \in \{1, \ldots, p-1\}$, remove the original edge, and identify u with u^*, v with v^*. Call the new graph G^*. By Proposition 9, each original edge with weight $w(e)$ ($w(e) \notin \{0,1\}$) contributes a factor of k when not selected in the perfect matching, and $k+1 = k \cdot w(e)$ when selected in the perfect matching. Combining with Theorem 3, we have

$$\mathrm{pm}(G^*) = \left(\prod_{\substack{e\in E(G_{\boldsymbol{X}}) \\ w(e)\notin\{0,1\}}} (w(e)-1)^{-1} \right) \cdot \mathrm{pm}(G_{\boldsymbol{X}}) = \left(\prod_{\substack{e\in E(G_{\boldsymbol{X}}) \\ w(e)\notin\{0,1\}}} (w(e)-1)^{-1} \right) \cdot \det(\boldsymbol{X}).$$

The factor in the above equation is non-zero modulo p. The number of vertices of the new graph G^* blows up by a factor of at most $3p$. It takes time $O(|V(G)| \cdot p)$ to construct the graph G^* and compute the factor. This turns any algorithm that computes $\mathrm{pm}(G^*) \bmod p$ in time $O(n^{\omega/2-\epsilon})$ into an algorithm that computes $\det(\boldsymbol{X})$ in time $O(m^{\omega-2\epsilon})$, a contradiction. ☐

References

[Bac21] Miriam Backens. A full dichotomy for Holant$^{\mathbf{c}}$, inspired by quantum computation. *SIAM J. Comput.*, 50(6):1739–1799, 2021. 1

[Bax82] Rodney J. Baxter. *Exactly Solved Models in Statistical Mechanics.* Academic press limited London, 1982. 1

[BCK+26] Cornelius Brand, Radu Curticapean, Petteri Kaski, Baitian Li, Ian Orzel, Tim Seppelt, and Jiaheng Wang. Beyond bilinear complexity: What works and what breaks with many modes? *Electron. Colloquium Comput. Complex.*, TR26-025, 2026. To appear in CCC 2026. 4

[Ber84] Stuart J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Inf. Process. Lett.*, 18(3):147–150, 1984. 3

[BH74] James R. Bunch and John E. Hopcroft. Triangular factorization and inversion by fast matrix multiplication. *Math. Comp.*, 28:231–236, 1974. 5

[BS83] Walter Baur and Volker Strassen. The complexity of partial derivatives. *Theor. Comput. Sci.*, 22:317–330, 1983. 2, 6

[Bür24] Peter Bürgisser. Completeness classes in algebraic complexity theory. *CoRR*, abs/2406.06217, 2024. 3

[CC07] Jin-yi Cai and Vinay Choudhary. Some results on matchgates and holographic algorithms. *Int. J. Softw. Informatics*, 1(1):3–36, 2007. 5

[CC17] Jin-Yi Cai and Xi Chen. *Complexity dichotomies for counting problems. Vol. 1.* Cambridge University Press, Cambridge, 2017. Boolean domain. 1

[CCL09] Jin-yi Cai, Vinay Choudhary, and Pinyan Lu. On the theory of matchgate computations. *Theory Comput. Syst.*, 45(1):108–132, 2009. 5

[CF22] Jin-Yi Cai and Zhiguo Fu. Holographic algorithm with matchgates is universal for planar #csp over boolean domain. *SIAM J. Comput.*, 51(2):17–50, 2022. 1

[CFGW22] Jin-Yi Cai, Zhiguo Fu, Heng Guo, and Tyson Williams. FKT is not universal - A planar Holant dichotomy for symmetric constraints. *Theory Comput. Syst.*, 66(1):143–308, 2022. 1

[CG14] Jin-Yi Cai and Aaron Gorenstein. Matchgates revisited. *Theory Comput.*, 10:167–197, 2014. 5

[CL11] Jin-yi Cai and Pinyan Lu. Holographic algorithms: From art to science. *J. Comput. Syst. Sci.*, 77(1):41–61, 2011. 1

[CLS+24] Matthias Christandl, Vladimir Lysikov, Vincent Steffan, Albert H. Werner, and Freek Witteveen. The resource theory of tensor networks. *Quantum*, 8:1560, December 2024. 3

[CLX20] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Dichotomy for Holant* problems on the boolean domain. *Theory Comput. Syst.*, 64(8):1362–1391, 2020. 1

[CX22] Radu Curticapean and Mingji Xia. Parameterizing the permanent: Hardness for fixed excluded minors. In Karl Bringmann and Timothy M. Chan, editors, *5th Symposium on Simplicity in Algorithms, SOSA@SODA 2022, Virtual Conference, January 10-11, 2022*, pages 297–307. SIAM, 2022. 5

[Edm65] Jack Edmonds. Paths, trees, and flowers. *Canadian J. Math.*, 17:449–467, 1965. 1

[EV15] G. Evenbly and G. Vidal. Tensor network renormalization. *Phys. Rev. Lett.*, 115:180405, Oct 2015. 3

[FKL07] Uffe Flarup, Pascal Koiran, and Laurent Lyaudet. On the expressive power of planar perfect matching and permanents of bounded treewidth matrices. In Takeshi Tokuyama, editor, *Algorithms and Computation, 18th International Symposium, ISAAC 2007, Sendai, Japan, December 17-19, 2007, Proceedings*, volume 4835 of *Lecture Notes in Computer Science*, pages 124–136. Springer, 2007. 3

[Geo73] Alan George. Nested dissection of a regular finite element mesh. *SIAM J. Numer. Anal.*, 10:345–363, 1973. 2

[HL16] Sangxia Huang and Pinyan Lu. A dichotomy for real weighted Holant problems. *Comput. Complex.*, 25(1):255–304, 2016. 1

[HPM+19] William Huggins, Piyush Patil, Bradley Mitchell, K Birgitta Whaley, and E Miles Stoudenmire. Towards quantum machine learning with tensor networks. *Quantum Science and Technology*, 4(2):024001, Jan 2019. 3

[IP01] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. 2

[JVW90] F. Jaeger, D. L. Vertigan, and D. J. A. Welsh. On the computational complexity of the Jones and Tutte polynomials. *Math. Proc. Cambridge Philos. Soc.*, 108(1):35–53, 1990. 2

[Kas63] P. W. Kasteleyn. Dimer statistics and phase transitions. *J. Mathematical Phys.*, 4:287–293, 1963. 1

[KK08] Erich L. Kaltofen and Pascal Koiran. Expressing a fraction of two determinants as a determinant. In J. Rafael Sendra and Laureano González-Vega, editors, *Symbolic and Algebraic Computation, International Symposium, ISSAC 2008, Linz/Hagenberg, Austria, July 20-23, 2008, Proceedings*, pages 141–146. ACM, 2008. 3

[LRT79] Richard J. Lipton, Donald J. Rose, and Robert Endre Tarjan. Generalized nested dissection. *SIAM J. Numer. Anal.*, 16(2):346–358, 1979. 2

[MP08] Guillaume Malod and Natacha Portier. Characterizing Valiant's algebraic complexity classes. *J. Complex.*, 24(1):16–38, 2008. 3

[MS06] Marcin Mucha and Piotr Sankowski. Maximum matchings in planar graphs via gaussian elimination. *Algorithmica*, 45(1):3–20, 2006. 2

[MV97] Meena Mahajan and V. Vinay. Determinant: Combinatorics, algorithms, and complexity. *Chic. J. Theor. Comput. Sci.*, 1997, 1997. 3

[Sam42] P. A. Samuelson. A method of determining explicitly the coefficients of the characteristic equation. *Ann. Math. Statistics*, 13:424–429, 1942. 3

[SC20] Shuai Shao and Jin-Yi Cai. A dichotomy for real boolean Holant problems. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1091–1102. IEEE, 2020. 1

[Sch81] Arnold Schönhage. Partial and total matrix multiplication. *SIAM J. Comput.*, 10(3):434–455, 1981. 6

[Str69] Volker Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13:354–356, 1969. 5

[TF61] H. N. V. Temperley and Michael E. Fisher. Dimer problem in statistical mechanics—an exact result. *Philos. Mag. (8)*, 6:1061–1063, 1961. 1

[Tod92] Seinosuke Toda. Classes of arithmetic circuits capturing the complexity of computing the determinant. *IEICE Transactions on Information and Systems*, 75(1):116–124, 1992. 3

[Val79a] Leslie G. Valiant. Completeness classes in algebra. In Michael J. Fischer, Richard A. DeMillo, Nancy A. Lynch, Walter A. Burkhard, and Alfred V. Aho, editors, *Proceedings of the 11h Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA*, pages 249–261. ACM, 1979. 3

[Val79b] Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 8(3):410–421, 1979. 1

[Val02a] Leslie G. Valiant. Expressiveness of matchgates. *Theor. Comput. Sci.*, 289(1):457–471, 2002. 5

[Val02b] Leslie G. Valiant. Quantum circuits that can be simulated classically in polynomial time. *SIAM J. Comput.*, 31(4):1229–1254, 2002. 5

[Val06] Leslie G. Valiant. Accidental algorithms. In *47th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2006, Berkeley, California, USA, October 21-24, 2006, Proceedings*, pages 509–517. IEEE Computer Society, 2006. 1

[Val08] Leslie G. Valiant. Holographic algorithms. *SIAM J. Comput.*, 37(5):1565–1594, 2008. 1

[VW18] Virginia Vassilevska Williams and R. Ryan Williams. Subcubic equivalences between path, matrix, and triangle problems. *J. ACM*, 65(5):27:1–27:38, 2018. 2

[Wel93] Dominic Welsh. *Complexity: Knots, Colourings and Countings*. London Mathematical Society Lecture Note Series. Cambridge University Press, 1993. 1

[Wil97] David Bruce Wilson. Determinant algorithms for random planar structures. In Michael E. Saks, editor, *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, 5-7 January 1997, New Orleans, Louisiana, USA*, pages 258–267. ACM/SIAM, 1997. 2

[Wil18] R. Ryan Williams. Faster all-pairs shortest paths via circuit complexity. *SIAM J. Comput.*, 47(5):1965–1985, 2018. 2

[Yus08] Raphael Yuster. Matrix sparsification for rank and determinant computations via nested dissection. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, Philadelphia, PA, USA, October 25-28, 2008*, pages 137–145. IEEE Computer Society, 2008. 2

[YZ07] Raphael Yuster and Uri Zwick. Maximum matching in graphs with an excluded minor. In Nikhil Bansal, Kirk Pruhs, and Clifford Stein, editors, *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 108–117. SIAM, 2007. 2