A Full Complexity Dichotomy for Immanant Families

RADU CURTICAPEAN*, IT University of Copenhagen, Denmark

Given an integer $n \geq 1$ and an irreducible character χ_λ of S_n for some partition λ of n, the immanant $\mathrm{imm}_\lambda : \mathbb{C}^{n\times n} \to \mathbb{C}$ maps matrices $A \in \mathbb{C}^{n\times n}$ to

$$\mathrm{imm}_\lambda(A) = \sum_{\pi \in S_n} \chi_\lambda(\pi) \prod_{i=1}^{n} A_{i,\pi(i)}.$$

Important special cases include the *determinant* and *permanent*, which are obtained from the *sign* and *trivial* character, respectively.

It is known that immanants can be evaluated in polynomial time for characters that are "close" to the sign character: Given a partition λ of n with s parts, let $b(\lambda) := n - s$ count the boxes to the right of the first column in the Young diagram of λ. For a family of partitions Λ, let $b(\Lambda) := \max_{\lambda \in \Lambda} b(\lambda)$ and write $\mathrm{Imm}(\Lambda)$ for the problem of evaluating $\mathrm{imm}_\lambda(A)$ on input A and $\lambda \in \Lambda$. On the positive side, if $b(\Lambda) < \infty$, then $\mathrm{Imm}(\Lambda)$ is known to be polynomial-time computable. This subsumes the case of the determinant. Conversely, if $b(\Lambda) = \infty$, then previously known hardness results suggest that $\mathrm{Imm}(\Lambda)$ cannot be solved in polynomial time. However, these results only address certain restricted classes of families Λ.

In this paper, we show that the assumption FPT ≠ W[1] from parameterized complexity rules out polynomial-time algorithms for $\mathrm{Imm}(\Lambda)$ for any computationally reasonable family of partitions Λ with $b(\Lambda) = \infty$. We give an analogous result in algebraic complexity under the assumption VFPT ≠ VW[1]. Furthermore, if $b(\lambda)$ even grows polynomially in Λ, we show that $\mathrm{Imm}(\Lambda)$ is hard for #P and VNP. This concludes a series of partial results on the complexity of immanants obtained over the last 35 years.

1 INTRODUCTION

The determinant and permanent of an $n \times n$ matrix $X = (x_{i,j})$ can be defined by the sum-product formulas

$$\det(X) = \sum_{\pi \in S_n} \mathrm{sgn}(\pi) \prod_{i=1}^{n} x_{i,\pi(i)},$$

$$\mathrm{per}(X) = \sum_{\pi \in S_n} \prod_{i=1}^{n} x_{i,\pi(i)}.$$

While determinants admit polynomial-size circuits and can be evaluated in polynomial time, only exponential-size circuits and exponential-time algorithms are known for permanents. Valiant [38] underpinned this divide by proving that evaluating permanents is #P-hard: Any polynomial-time algorithm for this problem would entail a polynomial-time algorithm for counting (and thus deciding the existence of) satisfying assignments to Boolean formulas, thereby collapsing P and NP. With the VNP-completeness of the permanent family, Valiant [37] showed an analogous statement in algebraic complexity theory.

Unconditional lower bounds for the complexity of permanents however remain elusive, with only a quadratic lower bound on the determinantal complexity of permanents known [12, 28]. That is, expressing the permanent of an $n \times n$ matrix X as the determinant of an $m \times m$ matrix (whose entries are linear forms in the entries of X) is known to require $m = \Omega(n^2)$. One of the core objectives in algebraic complexity theory lies in proving that m must grow super-polynomially [8, 11, 37], and this can be viewed as an algebraic version of the P ≠ NP problem.

The family of immanants. To gain a better understanding of the relationship between determinants and permanents, it may help to recognize them as part of a larger family: The *immanants* are matrix forms that are arranged on a spectrum

*Also with Basic Algorithms Research Copenhagen.

in which the determinant and permanent represent extreme cases. These forms were studied by Schur [32, 33] in the context of group character theory, and Littlewood and Richardson later explicitly introduced them as immanants [24].

Given any *class function* $f : S_n \to \mathbb{C}$, that is, a function of permutations that depends only on the (multiset of) cycle lengths of the input permutation, the immanant $\mathrm{imm}_f : \mathbb{C}^{n\times n} \to \mathbb{C}$ is defined by replacing the permutation sign $\mathrm{sgn}(\pi)$ in the determinant expansion with $f(\pi)$:

$$\mathrm{imm}_f(X) = \sum_{\pi \in S_n} f(\pi) \prod_{i=1}^{n} x_{i,\pi(i)}.$$

In the literature, immanants are typically defined by requiring f to be an *irreducible character* of S_n, i.e., an element from a particular basis for the vector space of class functions.[1] General f-immanants can then be expressed as linear combinations of such immanants. Two extremal examples of irreducible characters are the *trivial* character $\mathbf{1} : S_n \to \{1\}$ and the *sign* character $\mathrm{sgn} : S_n \to \{-1, 1\}$, which induce

$$\det(X) = \mathrm{imm}_{\mathrm{sgn}}(X),$$
$$\mathrm{per}(X) = \mathrm{imm}_{\mathbf{1}}(X).$$

The irreducible characters of S_n correspond naturally to partitions of n, as outlined in Section 3. To see the existence of such a correspondence, note that the dimension of the space of class functions on S_n is the number of different cycle length formats of n-permutations, that is, different partitions of the integer n. For now, let us remark that the refinement-wise minimal and maximal partitions $(1, \ldots, 1)$ and (n) naturally correspond to the sign and trivial character, respectively. As another example, we have

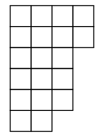
$$\chi_{(2,1,\ldots,1)}(\pi) = \mathrm{sgn}(\pi) \cdot (\#\{\text{fixed points of } \pi\} - 1). \quad (1)$$

Abbreviating $\mathrm{imm}_\lambda = \mathrm{imm}_{\chi_\lambda}$, we have $\det = \mathrm{imm}_{(1,\ldots,1)}$ and $\mathrm{per} = \mathrm{imm}_{(n)}$. Likewise, the immanant $\mathrm{imm}_{(2,1,\ldots,1)}$ sums over row-column permutations of a matrix with weights as given in (1). For a more applied example, it is known that the number of Hamiltonian cycles in a directed n-vertex graph G (that is, the immanant associated with the indicator function for cyclic permutations, evaluated on the adjacency matrix of G) is a linear combination of the *hook* immanants $\mathrm{imm}_{(r,1^{n-r})}$ for $1 \leq r \leq n$.

Beyond their origins in group character theory, immanants have been applied in combinatorial chemistry [13] and linear optics [35], and they feature in (conjectured) inequalities in matrix analysis [34].

The complexity of immanants. Any character immanant of an $n \times n$ matrix can be evaluated in $n! \cdot n^{O(1)}$ time by brute-force, or in $2^{n+o(n)}$ time by a variant of the Bellman–Held–Karp dynamic programming approach for Hamiltonian cycles. (The relevant irreducible characters of S_n can be evaluated in $2^{o(n)}$ time by the Murnaghan-Nakayama rule, described in Section 3, together with dynamic programming.) For some immanants however, this exponential running time is far from optimal: The determinant is a first such example, and more generally, Hartmann [23] gave an algorithm for evaluating imm_λ in $O(n^{6b(\lambda)+4})$ time, where $b(\lambda) := n - s$ for a partition $\lambda = (\lambda_1, \ldots, \lambda_s)$ with s parts. In visual terms, the quantity $b(\lambda)$ counts the boxes to the right of the first column in the Young diagram of λ, which is a left-aligned shape whose i-th row contains λ_i boxes, when λ is ordered non-increasingly:

[1] The resulting immanants are sometimes also called *character* immanants, as opposed to other types of immanants, such as the Kazhdan-Lusztig immanants [30].



$\lambda = (4, 4, 3, 3, 3, 2)$ with $b(\lambda) = 13$

Barvinok [1] and Bürgisser [10] later gave $O(n^2 d_\lambda^4)$ and $O(n^2 s_\lambda d_\lambda)$ time algorithms, where s_λ and d_λ denote the numbers of *standard* and *semi-standard* tableaux of shape λ.[2] These algorithms give better running times in the exponential-time regime, but they do not identify new polynomial-time solvable immanants. One is therefore naturally led to wonder whether $b(\lambda)$ is indeed the determining parameter for the complexity of immanants. To investigate this formally, we consider *families* of partitions Λ and define $\mathrm{Imm}(\Lambda)$ as the problem of evaluating $\mathrm{imm}_\lambda(A)$ on input a matrix A and a partition $\lambda \in \Lambda$. As discussed above, the problem $\mathrm{Imm}(\Lambda)$ is polynomial-time solvable if the quantity

$$b(\Lambda) := \max_{\lambda \in \Lambda} b(\lambda)$$

is finite. On the other hand, for various families Λ with unbounded $b(\Lambda)$, the problem $\mathrm{Imm}(\Lambda)$ is indeed known to be hard for the counting complexity class #P and its algebraic analog VNP:

- Bürgisser [9] showed VNP-completeness and #P-hardness of $\mathrm{Imm}(\Lambda)$ for any family Λ of hook partitions $(t(n), 1^{n-t(n)})$, provided that $t = \Omega(n^\alpha)$ with $\alpha > 0$ can be computed in polynomial time. A similar result appears in Hartmann's work [23].
- In the same paper, Bürgisser showed similar hardness results for families of rectangular partitions of polynomial width. (The width is the largest entry in the partition.)

In his 2000 monograph [8], Bürgisser conjectures that $\mathrm{Imm}(\Lambda)$ is hard for *any* reasonable family Λ of polynomial width. He also asks about the complexity status of partitions of width 2, and overall deems the complexity of immanants to be "still full of mysteries". Some of these mysteries have since been resolved:

- In 2003, Brylinski and Brylinski [7] showed VNP-completeness for any family of partitions Λ with a gap of width $\Omega(n^\alpha)$ for $\alpha > 0$. Here, a gap is the difference between two consecutive rows.
- In 2013, Mertens and Moore [27] proved #P-hardness for the family Λ of *all* partitions of width 2, that is, the partitions containing only entries 1 and 2. They also proved $\oplus$P-hardness for the more restricted family of partitions containing only the entry 2.
- In the same year, de Rugy-Altherre [20] gave a dichotomy for partition families Λ of constant width and polynomial growth of $b(\lambda)$, confirming for such families that boundedness of $b(\Lambda)$ indeed determines the complexity of $\mathrm{Imm}(\Lambda)$.

However, an exhaustive complexity classification of $\mathrm{Imm}(\Lambda)$ for general partition families Λ still remained open, even 35 years after Hartmann's initial paper [23] and despite several appearances as an open problem [20, 27], also in a monograph [8]. In fact, even very special cases like $\mathrm{Imm}(\Lambda)$ for the staircase partitions $(k, k-1, \ldots, 1)$ remained unresolved [20].

[2]Given a partition λ of n, a standard tableau of shape λ is an assignment of the numbers $1, \ldots, n$ to the boxes in the Young diagram of λ such that all rows and columns are strictly increasing. In a semi-standard tableau, rows are only required to be non-decreasing.

1.1 Our Results

We classify the complexity of the problems $\mathrm{Imm}(\Lambda)$ for partition families Λ satisfying natural computability and density conditions that are satisfied by all families studied in the literature. Under the assumption $\mathsf{FPT} \neq \#\mathsf{W}[1]$ from parameterized complexity [21], we confirm that $\mathrm{Imm}(\Lambda)$ is polynomial-time solvable iff $b(\Lambda)$ is unbounded. An algebraic analogue holds under the assumption $\mathsf{VFPT} \neq \mathsf{VW}[1]$ introduced by Bläser and Engels [6]. (Please consider Section 2.2 for a brief introduction to the relevant complexity classes.) Our classification holds even if $b(\lambda)$ only grows sub-polynomially in Λ, which allows us to address families such as

$$\Lambda_{\log} = \{(\lceil \log n \rceil, 1^n) \mid n \in \mathbb{N}\}. \tag{2}$$

Note that $\mathrm{Imm}(\Lambda_{\log})$ can be solved in $n^{O(\log n)}$ time by the $n^{O(b(\lambda))}$ time algorithms discussed before, which likely prevents hardness for $\#\mathsf{P}$ or VNP. At the same time, a polynomial-time algorithm seems unlikely. Thus, partition families like $\mathrm{Imm}(\Lambda_{\log})$ fall into the "blind spot" of classical dichotomies.

Our sanity requirements on Λ are encapsulated as follows: We say that Λ *supports growth* $g : \mathbb{N} \to \mathbb{N}$ if every $n \in \mathbb{N}$ admits a partition $\lambda^{(n)} \in \Lambda$ with $b(\lambda^{(n)}) \geq g(n)$ and total size $\Theta(n)$. This ensures that Λ is dense enough and that Λ supplies sufficiently many boxes both in the first column *and* to the right of it. We may also require that $\lambda^{(n)}$ can be computed in polynomial time on input $n \in \mathbb{N}$ and then say that Λ *computationally* supports growth g. This condition is not required for the algebraic completeness results.

Example. The family of staircase partitions $(n, n-1, \ldots, 1)$ for $n \in \mathbb{N}$ supports growth $\Omega(n)$. The partition families $(\lceil \log n \rceil, 1^n)$ and $(n, 1^{2^n})$ for $n \in \mathbb{N}$ support growth $\Omega(\log n)$, even though the second family is exponentially sparse. On the other hand, partition families whose sizes grow *doubly* exponentially do not support any growth by our definition. It might still be possible to address such families via "infinitely often" versions of $\#\mathsf{P}$ or VNP, but we currently see no added value in doing so.

In the polynomial growth regime for $b(\lambda)$, we obtain classical $\#\mathsf{P}$-hardness and VNP-completeness results. As a bonus, we also obtain the expected quantitative lower bounds under the exponential-time hypothesis $\#\mathsf{ETH}$, which postulates that counting satisfying assignments to n-variable 3-CNFs takes $\exp(\Omega(n))$ time.

Theorem 1. *For any family of partitions Λ:*

- *If $b(\Lambda) < \infty$, then $\mathrm{Imm}(\Lambda) \in \mathsf{FP}$ and $\mathrm{Imm}(\Lambda) \in \mathsf{VP}$.*
- *Otherwise, if Λ supports growth $\Omega(n^\alpha)$ for some $\alpha > 0$, then $\mathrm{Imm}(\Lambda)$ is VNP-complete. If Λ computationally supports growth $\Omega(n^\alpha)$, then $\mathrm{Imm}(\Lambda)$ is $\#\mathsf{P}$-hard and admits no $\exp(o(n^\alpha))$ time algorithm unless $\#\mathsf{ETH}$ fails.*

Theorem 1 subsumes all known VNP-hardness and $\#\mathsf{P}$-hardness results for immanant families, confirms the conjecture from [27], and settles the case of staircases.

Using parameterized complexity theory, we also address the sub-polynomial growth regime for $b(\lambda)$. To this end, we consider *parameterized problems*, whose instances (x, k) come with a *parameter* k. The corresponding objects in the algebraic setting are *parameterized polynomial families* $(p_{n,k})$, where the second index k is a parameter. A parameterized problem (or polynomial family) is said to be *fixed-parameter tractable* if it can be solved in $f(k) \cdot n^{O(1)}$ time with $n = |x|$ (or admits circuits of that size) for some computable function f. The problem (or polynomial family) is then said to be contained in FPT (or VFPT). The classes $\#\mathsf{W}[1] \supseteq \mathsf{FPT}$ (and $\mathsf{VW}[1] \supseteq \mathsf{VFPT}$) contain problems (and polynomial families) that are believed not to be fixed-parameter tractable.

Theorem 2. *For any family of partitions* Λ*:*

- *If* $b(\Lambda) < \infty$*, then* $\mathrm{Imm}(\Lambda) \in \mathsf{FP}$ *and* $\mathrm{Imm}(\Lambda) \in \mathsf{VP}$*.*
- *Otherwise, if* Λ *supports growth* $g \in \omega(1)$*, then* $\mathrm{Imm}(\Lambda) \notin \mathsf{VP}$ *unless* $\mathsf{VFPT} = \mathsf{VW}[1]$*. If* Λ *computationally supports growth* g*, then* $\mathrm{Imm}(\Lambda) \notin \mathsf{FP}$ *unless* $\mathsf{FPT} = \#\mathsf{W}[1]$*.*

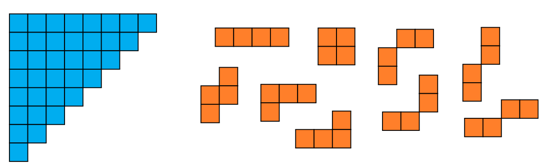
Note that we do not prove $\mathrm{Imm}(\Lambda)$ to be *hard* for #W[1] or VW[1] as a parameterized problem with parameter $b(\lambda)$, even though this might seem natural. Indeed, problems like $\mathrm{Imm}(\Lambda_{\log})$ are trivially fixed-parameter tractable in the parameter $b(\lambda)$. We only show that *polynomial-time* algorithms for $\mathrm{Imm}(\Lambda)$ would render #W[1]-hard problems fixed-parameter tractable.

1.2 Proof Outline

We establish Theorems 1 and 2 by reduction from #Match, the problem of counting k-matchings in bipartite graphs H. When parameterized by k, this problem is #W[1]-complete [4, 15, 18, 19], with an analogous statement in the algebraic setting [6]. When k grows polynomially in $|V(H)|$, counting k-matchings is complete for #P and VNP by a trivial reduction from the permanent [38].

To reduce counting matchings to immanants, we proceed in three stages: First, we identify two types of "exploitable resources" in partitions, then we show how to exploit them for a reduction, and finally we wrap the proof up in complexity-theoretic terms.

Extracting resources (Section 4). Our construction relies on two types of resources that can supplied by a given partition: A large *staircase* or a large number of *non-vanishing tetrominos*.



To define these objects, consider successively "peeling" dominos □□ and ⊟ from λ, that is, removing them from the south-eastern border of λ while ensuring that the shape obtained after each step has non-increasing row lengths. After peeling the maximum number of dominos this way, we reach some (possibly empty) staircase μ, which is easily seen to be unique. The *domino number* $d(\lambda)$ is this maximum number of removable dominos, and μ is the *staircase of* λ; we write $w(\lambda)$ for its width.

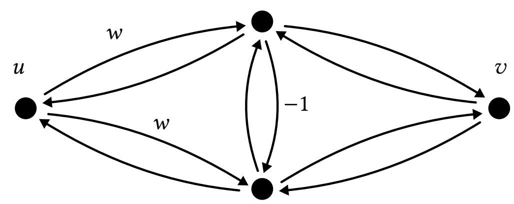
Now consider peeling *two* dominos from a partition λ. Some of the shapes that can arise this way are shown above in orange; we call these shapes "non-vanishing tetrominos" for reasons that will become evident in the proof. For the four corner-connected domino pairs, we adopt the convention that their two dominos must be peeled successively from disjoint rows and columns of λ. The *non-vanishing tetromino number* $s(\lambda)$ then is the maximum number of non-vanishing tetrominos that can be peeled from λ. Note that this number is 0 for the determinant-inducing partition $(1, \ldots, 1)$ and $\lfloor n/4 \rfloor$ for the permanent-inducing partition (n).

In Section 4, we establish a "win-win situation": For any partition λ with large $b(\lambda)$, either the staircase width or the non-vanishing tetromino number is large. That is, at least one of $w(\lambda) \in \Omega(\sqrt{b(\lambda)})$ or $s(\lambda) \in \Omega(b(\lambda))$ must hold.

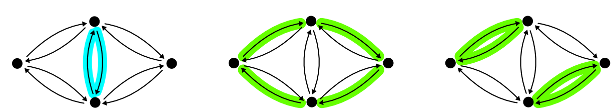
Exploiting resources (Sections 5 and 6). Next, we outline how to exploit staircases and non-vanishing tetrominos in a partition λ for reductions from counting k-matchings to evaluating imm_λ. Throughout this paper, the immanant of a directed graph G refers to the immanant of its adjacency matrix A, and we view immanants as character-weighted sums over the cycle covers of digraphs.

Given an n-vertex graph H and $k \in \mathbb{N}$, we construct a digraph G such that $\mathrm{imm}_\lambda(G)$ counts the k-matchings in H up to a constant factor $c_{\lambda,k}$ that can be computed in polynomial time. In a second step, we show that the constant factor $c_{\lambda,k}$ is non-zero if λ supplies enough resources. Both the construction of G and the handling of $c_{\lambda,k}$ differ for staircases and tetrominos, as outlined below.

Non-vanishing tetrominos (Section 6). Each non-vanishing tetromino peeled from λ enables a particular *edge gadget*: To count the k-matchings in a graph H, we replace each edge $uv \in E(H)$ by the gadget shown below; the weight w of uv appears on two edges of the gadget. Together with additional constructions detailed in Section 6, this results in a directed graph G.



The edge gadget effectively constrains the set of cycle covers counted by the immanant, as undesired cycle covers cancel out in pairs of opposite signs. In the remaining cycle covers of G, each gadget is either in the *passive state* (shown below in cyan) or in one of four *active states* (two are shown below in green, two more are symmetric versions thereof).



This allows us to simulate matchings M in H via cycle covers in G: We interpret active gadgets as matching edges $e \in M$ and passive gadgets as edges $e \in E(H) \setminus M$. Intuitively speaking, each active gadget "uses up" one non-vanishing tetromino of λ, while passive gadgets only require a domino. Overall, if we can peel $O(k)$ non-vanishing tetrominos and some number of dominos from λ, then imm_λ can be used to count k-matchings in H.

Large staircase (Section 5). If the staircase μ of λ contains $\Omega(k)$ boxes, then properties of the staircase character χ_μ enable an ad-hoc reduction from counting k-matchings in bipartite graphs to the λ-immanant. More specifically, we observe and use that cycle covers with even cycles vanish in staircase characters χ_μ. After discarding irrelevant dominos from λ, we can then use this fact together with a particular graph construction to compute a sum over cycle covers with one *particular fixed* cycle length format by reduction to imm_λ. The fact that staircase characters allow us to avoid even-length cycles may also have algorithmic applications.

Wrap-up (Section 7). For a streamlined presentation, the two reductions above are encapsulated as formulas relating the number of k-matchings in a graph H with the immanant of a digraph G constructed from H. In Section 7, we add the necessary "wrapper code" to obtain the (parameterized and polynomial-time, algebraic and computational) reductions leading to Theorems 1 and 2.

1.3 Proof Highlights

Our arguments rely on making non-vanishing tetrominos and staircases come together in just the right way. This requires some technical effort. Three specific highlights can be pointed out in the proof strategy.

Firstly, in the tetromino-based reduction, the particular form of active and passive states in edge gadgets ensures that we only need to understand character values $\chi_\lambda(\rho)$ on cycle formats ρ with cycle lengths 1, 2, and 4. This allows us to establish the tetromino/staircase win-win argument, and it sidesteps more involved representation-theoretic arguments that occur in related works. It should be noted that *equality* and *exclusive-or* gadgets that enforce consistency properties of cycle covers are common in algebraic and counting complexity, dating back to Valiant [37, 38]. The particular idea of repurposing an equality gadget into an *edge* gadget was also already used in the author's very first paper with Bläser [3].

Secondly, parameterized complexity assumptions allow us to handle cases that cannot be addressed in classical frameworks, such as the family $\mathrm{Imm}(\Lambda_{\log})$ discussed before. By basing hardness on the assumptions $\mathsf{FPT} \neq \#\mathsf{W}[1]$ and $\mathsf{VFPT} \neq \mathsf{VW}[1]$, we can still argue about such families.

Finally, two inconspicuous but crucial proof steps (Lemma 19 and Fact 25) rely on cute arguments involving "dominos on chessboards" that one would rather expect in the context of recreational mathematics. This can again be credited to the edge gadget, as such arguments would likely fail for cycle lengths other than 2 and 4.

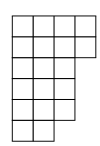
2 PRELIMINARIES

We start with basic definitions for graphs and partitions in Section 2.1. In Sections 2.2 and 2.3, we present complexity-theoretic preliminaries used in this paper. An introduction to the relevant character theory of symmetric groups can be found in Section 3.

2.1 Basic Notions

Graphs. We will consider undirected graphs (when counting matchings) and directed graphs (when evaluating immanants). Both graph types may feature (indeterminate) edge-weights, and directed graphs may feature self-loops. For a graph G with adjacency matrix A, we write $\mathrm{imm}(G)$ instead of $\mathrm{imm}(A)$ and view the immanant as a sum over cycle covers: A **cycle cover** in G is an edge-set $C \subseteq E(G)$ such that each vertex has exactly one incoming and one outgoing edge in C. Directed 2-cycles will also be called **digons**.

Partitions. A **partition** of a positive integer $n \in \mathbb{N}$ is a multi-set λ of positive integers such that $\sum_{i \in \lambda} i = n$. Its elements are called **parts**, and we write $\lambda \vdash n$ to indicate that λ is a partition of n, and we also write $n = \|\lambda\|$. Several notations will be used for partitions:

sorted tuple	compact notation	Young diagram
$(4, 4, 3, 3, 3, 2)$	$(4^2, 3^3, 2^1)$	

The **Young diagram** of a partition $\lambda = (\lambda_1, \ldots, \lambda_s)$ is a left-aligned shape consisting of λ_i boxes in row i. We define the **gap** δ_i of row i as $\delta_i := \lambda_i - \lambda_{i+1}$, where we consider $\lambda_{s+1} := 0$.

Given partitions $\lambda \vdash n$ and $\lambda' \vdash n'$, we sometimes abuse notation and write (λ, λ') for the partition $\lambda \cup \lambda'$ of $n + n'$. Sometimes we also specify an ordering for the elements in a partition: An **ordered partition** of n (also called **composition**) is a tuple of positive integers that sum to n. We will state explicitly when partitions are considered to be ordered.

Let us stress that any permutation π of n elements (and any cycle cover C of an n-vertex graph) naturally induces a partition $\lambda \vdash n$ through its cycle lengths, the **cycle format** $\rho(\pi)$ of π. For example, the identity permutation has cycle format $(1, \ldots, 1)$ and cyclic permutations have cycle format (n). We also say that a cycle cover is a ρ-cycle cover if its format is ρ.

2.2 Complexity Theory

We recall basic notions from complexity theory. For a more comprehensive overview, consider [17, 21, 38] for (parameterized) counting complexity and [6, 8, 11, 37] for (parameterized) algebraic complexity.

Counting complexity. We view functions $f : \{0, 1\}^* \to \mathbb{Q}$ as **counting problems**. For example, the problem #SAT maps (binary encodings of) Boolean formulas φ to the number of satisfying assignments in φ. Properly encoded, the permanent of rational-valued matrices is a counting problem. A counting problem is contained in FP if it can be solved in polynomial time.

A **polynomial-time Turing reduction** from a counting problem #A to another counting problem #B is a polynomial-time algorithm that solves #A with an oracle for #B. We say that #B is #P**-hard** (under Turing reductions) if #SAT admits a polynomial-time Turing reduction to #B. While more stringent reduction notions exist, they are not relevant for the purposes of this paper. Assuming FP ≠ #P, no #P-hard problem can be solved in polynomial time.

A **parameterized counting problem** features inputs (x, k) for $x \in \{0, 1\}^*$ and $k \in \mathbb{N}$. It is **fixed-parameter tractable** if it can be solved in time $f(k) \cdot |x|^{O(1)}$ for some computable function f, and we write FPT for the class of such problems. A **parameterized Turing reduction** from a parameterized counting problem #A to another parameterized problem #B is an algorithm that solves any instance (x, k) for #A in time $f(k) \cdot |x|^{O(1)}$ with an oracle for #B that is only called on instances (y, k') with $k' \leq g(k)$. Here, both f and g are computable functions. We say that #B is #W[1]**-hard** if the problem of counting k-cliques in a graph admits a parameterized Turing reduction to #B. Assuming FPT ≠ #W[1], no #W[1]-hard problem is fixed-parameter tractable.

The (counting version of the) **exponential-time hypothesis** #ETH postulates that no $\exp(o(n))$ time algorithm solves #SAT on n-variable formulas. If a problem #A cannot be solved in $\exp(o(n))$ time, and it admits a polynomial-time Turing reduction to a problem #B such that every invoked oracle query has size $O(n^c)$ for $c \geq 0$, then #B cannot be solved in $\exp(o(n^{1/c}))$ time. Likewise, if a parameterized problem #A cannot be solved in $f(k) \cdot n^{o(k)}$ time and it admits a parameterized Turing reduction to a problem #B such that every invoked query (y, k') satisfies $k' \leq O(k)$, then #B also cannot be solved in $f(k) \cdot n^{o(k)}$ time. The hypothesis #ETH rules out an $f(k) \cdot n^{o(k)}$ time algorithm for counting k-cliques and thus implies FPT ≠ #W[1].

Algebraic complexity. In the algebraic setting, **p-families** play the role of counting problems: A sequence of multivariate polynomials $f = (f_1, f_2, \ldots)$ over some field is a p-family if, for all $n \in \mathbb{N}$, the degree and the number of variables of f_n are bounded by $n^{O(1)}$. For example, the sequences of determinants and permanents of $n \times n$ matrices with indeterminates are p-families. A p-family is contained in VP if it admits a polynomial-size arithmetic circuit.

A p-family f admits a **c-reduction** to another p-family g if there is an arithmetic circuit of polynomial size that computes each f_n with oracle gates for $g_1, \dots g_{n^{O(1)}}$. A p-family g is VNP-**hard** if the permanent family admits a c-reduction to g. Assuming VP $\neq$ VNP, no VNP-hard family is contained in VP.

A **parameterized p-family** is a family $f = (f_{n,k})_{n,k \in \mathbb{N}}$ with two indices such that $f_{n,k}$ uses $n^{O(1)}$ variables and has degree $(n+k)^{O(1)}$. A family f is contained in VFPT if it admits an arithmetic circuit of size $h(k) \cdot n^{O(1)}$ for some function h. An example for a parameterized p-family is given by the **partial permanents** $\mathrm{per}_{n,k} = \sum_{\pi \in S(n,k)} \prod_{i=1}^{n} x_{i,\pi(i)}$ for $n, k \in \mathbb{N}$, where $S(n, k)$ is the set of all permutations with $n - k$ fixed points. A parameterized family $f = (f_{n,k})_{n,k \in \mathbb{N}}$ admits a **parameterized c-reduction** to another family $g = (g_{n,k})_{n,k \in \mathbb{N}}$ if there is an arithmetic circuit of size $h(k) \cdot n^{O(1)}$ that computes $f_{n,k}$ with oracle gates for polynomials $g_{n',k'}$ satisfying $n' \leq h(k) \cdot n^{O(1)}$ and $k' \leq h(k)$ for some function h. We say that g is VW[1]-**hard** if the partial permanents admit a parameterized c-reduction to g. Assuming VFPT $\neq$ VW[1], no VW[1]-hard problem is contained in VFPT.

2.3 Counting Matchings

We prove hardness of immanants by reduction from problems related to counting matchings in graphs. A **matching** in an undirected graph H is a set $M \subseteq E(H)$ of pairwise disjoint edges. We write $\mathcal{M}_k(H)$ for the set of matchings with k edges in H. A matching M is a **perfect matching** if every vertex $v \in V(H)$ is contained in some edge of M. Given an n-vertex graph H with edge-weights $w : E(H) \to \mathbb{Q}$, we define

$$\#\mathrm{Match}(H, k) = \sum_{M \in \mathcal{M}_k(H)} \prod_{e \in M} w(e),$$
$$\#\mathrm{PerfMatch}(H) = \#\mathrm{Match}(H, |V(H)|/2).$$

Note that #PerfMatch(H) is only defined for graphs H with an even number of vertices.

Definition 3. The counting problem #PerfMatch asks to compute #PerfMatch(H) for *bipartite* H with edge-weights $w : E(H) \to \mathbb{Q}$.

Furthermore, for any fixed polynomial-time computable function $g : \mathbb{N} \to \mathbb{N}$, we define the counting problem $\#\mathrm{Match}^{(g)}$: Given a pair (H, k) consisting of an undirected bipartite graph H with edge-weights $w : E(H) \to \mathbb{Q}$ and a number $k \leq g(|V(H)|)$, compute #Match(H, k).

On the complete bipartite graphs $K_{n,n}$ with indeterminate edge-weights, #PerfMatch induces the p-family of permanent polynomials, and $\#\mathrm{Match}^{(g)}$ likewise induces a restriction of the partial permanent family with $\#\mathrm{Match}^{(g)}_{n,k} = 0$ for $k > g(n)$. Abusing notation, we call these p-families #PerfMatch and $\#\mathrm{Match}^{(g)}$ as well. Note that no graphs are given as inputs to these families; the numbers of k-matchings for given bipartite graphs H can be obtained by evaluating the polynomials at points whose non-zero coordinates encode the edges of H.

The hardness results for #PerfMatch and $\#\mathrm{Match}^{(g)}$ required in the remainder of the paper are either known in the literature or can be derived easily. We collect the relevant results below.

Theorem 4. *The following holds:*

1. *The problem* #PerfMatch *is* VNP-*complete and* #P-*hard and admits no* $2^{o(n)}$ *time algorithm under* #ETH*, even on bipartite graphs of maximum degree* 3.
2. *For any unbounded and polynomial-time computable function g, the problem* $\#\mathrm{Match}^{(g)}$ *is* #W[1]-*hard and* VW[1]-*complete.*

Proof. See [37, 38] for the #P-hardness and VNP-completeness of #PerfMatch and [16] for the lower bound under #ETH. The #W[1]-hardness of the cardinality-unrestricted problem #Match is shown in [19], and and VW[1]-completeness is shown in [6, Lemma 8.7].

For any polynomial-time computable function g, we give a parameterized reduction from #Match to #Match$^{(g)}$: Any instance (H, k) with an n-vertex graph and $k \leq g(n)$ can be solved directly with a call to #Match$^{(g)}$. On the other hand, if $k > g(n)$, then we have $n < g^{-1}(k)$, so we can count k-matchings in H by brute-force in $g'(k)$ time for some computable function g'. This satisfies the requirements of a parameterized reduction.

In the algebraic setting, we need not branch on $g(n)$ within a circuit, but instead hard-code into the circuit family realizing #Match whether to (i) perform a brute-force sum over matchings, or (ii) use the oracle gate for #Match$^{(g)}$. □

3 CHARACTERS OF THE SYMMETRIC GROUP

We give a minimal introduction to representations and characters of the symmetric group; this material is covered thoroughly in classical textbooks [22, 25, 26, 31]. For our purposes, a **representation** of S_n is a homomorphism f from S_n to the group of complex-valued invertible $t \times t$ matrices, for some $t \in \mathbb{N}$. Examples include the trivial representation that maps all of S_n to 1, the sign representation that maps permutations to their sign, and the permutation matrix representation that maps permutations to $n \times n$ permutation matrices.

A representation $f : S_n \to \mathrm{GL}_t(\mathbb{C})$ is **irreducible** if no proper subspace of $\mathbb{C}^t$ is invariant under all the transformations $f(\pi)$ for $\pi \in S_n$. Among the examples given before, this holds trivially for the trivial and sign representations. The permutation matrix representation however is not irreducible for $n > 1$, as every permutation matrix maps the 1-dimensional subspace of $\mathbb{C}^n$ that is spanned by $(1, \ldots, 1)$ to itself.

3.1 Characters

Characters condense essential information about representations $f : S_n \to \mathrm{GL}_t(\mathbb{C})$ into scalar-valued functions $\chi_f : S_n \to \mathbb{C}$. For us, they play the role of "generalized signs" in the sum-product definition of immanants.

Definition 5. The **character** of a representation f is the function $\chi_f : S_n \to \mathbb{C}$ that maps $\pi \in S_n$ to the trace of the matrix $f(\pi)$.

The trivial and sign representations coincide trivially with their characters. The character of the permutation matrix representation counts the fixed points of a permutation.

Characters of representations are **class functions**, which are functions $f : S_n \to \mathbb{C}$ that depend only on the cycle format of the input. These functions form a vector space by point-wise linear combinations, and a particularly useful basis for this space is given by the **irreducible characters**, which are the characters of irreducible representations. The set of irreducible characters corresponds bijectively to the partitions of n.

For S_2, the only irreducible characters are the trivial character $\chi_{(2)}$ and sign character $\chi_{(1,1)}$. There are five irreducible characters for S_4; their values $\chi_\lambda(\rho)$ are shown below as a character table.

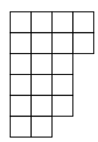
ρ

	$\chi_\lambda(\rho)$	(1^4)	$(2^1, 1^2)$	(2^2)	$(3^1, 1^1)$	(4^1)
	(4^1)	1	1	1	1	1
	$(3^1, 1^1)$	3	1	−1	0	−1
λ	(2^2)	2	0	2	−1	0
	$(2^1, 1^2)$	3	−1	−1	0	1
	(1^4)	1	−1	1	1	−1

In the next subsections, we describe the Murnaghan-Nakayama rule, a combinatorial method for calculating character values $\chi_\lambda(\rho)$. This rule was used before in works on immanants [20]. We also give a simple extension of the Murnaghan-Nakayama rule that applies to particular linear combinations of character values. To state these rules, we need to introduce several types of *tableaux*.

3.2 Skew Shapes and Tableaux

Recall that partitions λ can be described by Young diagrams, e.g.,

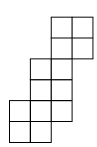


$(4^2, 3^3, 2^1)$.

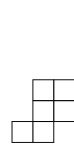
Given such a diagram, a tableau is obtained by writing numbers (or other objects) into the boxes, subject to some specified rules. The representation theory of S_n abounds in different types of tableaux—we introduce yet another such type, the *skew shape tableaux.*

Definition 6. Let λ, μ be partitions such that $\mu_i \leq \lambda_i$ for all rows i of λ. We say that μ is *contained* in λ and define the **skew shape** λ/μ by removing the diagram of μ from λ.

Consider the two examples below. The right example shows that skew shapes need not be connected; we call a skew shape **connected** if each pair of boxes can be reached by a path in the interior of the shape.



$(4^2, 3^3, 2^1)/(2^2, 1^2)$



$(4^2, 3^3, 2^1)/(3^3, 1^2)$

Figure 1 lists the connected skew shapes on 4 boxes. Any general skew shape on b boxes is obtained by choosing connected skew shapes with a total of b boxes and arranging them in disjoint rows and columns.

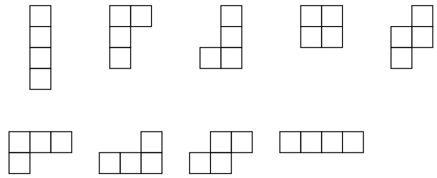


Fig. 1. All connected skew shapes on 4 boxes.

We will often peel skew shapes γ from other skew shapes λ/μ. This operation is used, e.g., in the Murnaghan-Nakayama rule. Our definition of this process allows for peeling different components of γ from different places.

Definition 7. A skew shape γ can be **peeled** from λ/μ if there is a partition λ' contained between μ and λ such that λ/λ' equals γ after deleting empty rows and columns from both λ/λ' and γ.

For example, in Figure 2, the green skew shape can be peeled from the partition λ; this holds even after removing empty rows and columns from the green shape. A skew shape *tableau* is obtained by successively peeling skew shapes from λ.

Definition 8. Let $\tilde{\Gamma} = (\Gamma_1, \dots, \Gamma_s)$ be such that Γ_i for $i \in [s]$ is a set of skew shapes on the same number n_i of boxes. Let λ be a partition of $n = \sum_i n_i$. A **skew shape tableau** of λ with format $\tilde{\Gamma}$ is obtained by successively peeling skew shapes $\gamma_1, \dots, \gamma_s$ with $\gamma_i \in \Gamma_i$ from λ and labeling γ_i with i in λ. We write $\mathcal{S}(\lambda, \tilde{\Gamma})$ for the set of such tableaux.

Three skew shape tableaux are shown in Figure 2. The boxes are *colored* rather than numbered. The third tableau is even a *border strip* tableau, as defined in the next subsection.

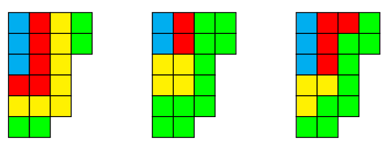


Fig. 2. Three skew shape tableaux.

3.3 An Extension of the Murnaghan-Nakayama Rule

The Murnaghan-Nakayama rule expresses the character value $\chi_\lambda(\rho)$ for partitions λ and ρ as a signed sum over particular skew shape tableaux of λ. The sign of a tableau is determined by the parity of odd-height shapes. In our later arguments, ρ will always be the format of a cycle cover.

Definition 9. A **border strip** is a connected skew shape not containing any 2×2 square. A **border strip tableau** of $\lambda \vdash n$ is a skew shape tableau consisting only of border strips. Given a partition λ and an ordered partition κ, we write $\mathcal{B}(\lambda, \kappa)$ for the set of border strip tableaux of λ in which the i-th shape has κ_i boxes.

The **height** $\mathrm{ht}(\gamma)$ of a border strip γ is the number of occupied rows in γ minus 1. Given a skew shape tableau T with skew shapes $\gamma_1, \dots, \gamma_s$, the **height sign** $\mathrm{ht}(T)$ is $\mathrm{ht}(T) = \prod_{i=1}^{s} (-1)^{\mathrm{ht}(\gamma_i)}$.

For the border strip tableau in the right part of Figure 2, the heights are 5 (green), 1 (yellow), 2 (red), and 2 (blue), resulting in an overall height sign of +1. We are now ready state the Murnaghan-Nakayama rule and refer to textbooks [22, 25] for proofs.

Theorem 10 (Murnaghan-Nakayama rule)**.** *For partitions λ and ρ and any ordering κ of ρ, we have*

$$\chi_\lambda(\rho) = \sum_{T \in \mathcal{B}(\lambda,\kappa)} \mathrm{ht}(T).$$

Remark 11. When invoking this rule, it makes sense to choose a useful ordering κ of ρ. For example, to show $\chi_{(5,4,3,2,1)}(1^3, 2^6) = 0$, we can either (i) sum over a rather large number of border strip tableaux to observe that their signs cancel, or (ii) reorder $\rho = (1^3, 2^6)$ to $\kappa = (2^6, 1^3)$ and realize directly that no border strip on 2 boxes can be peeled from the border of the staircase $(5, 4, 3, 2, 1)$.

More generally, this shows that $\chi_\mu(\rho) = 0$ whenever μ is a staircase and ρ is a partition that contains at least one even part.

This remark will be crucial for the staircase-based reduction in Section 5. For the tetromino-based reduction in Section 6, we extend Theorem 10 to character evaluations on products of partition sets.

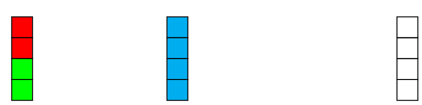
Definition 12. Let $F_1, \ldots, F_t$ be sets of partitions such that each set F_i collects partitions of the same integer d_i. The *partition product* $F_1 \times \ldots \times F_t$ is the multi-set consisting of the $\prod_i |F_i|$ partitions of $d_1 + \ldots + d_t$ obtained by choosing one partition from each set F_i and concatenating those t partitions.

We extend class functions f to partition multi-sets S by declaring $f(S) = \sum_{\rho \in S} f(\rho)$ and show how to calculate $\chi_\lambda(F_1 \times \ldots \times F_t)$ combinatorially by an extension of the Murnaghan-Nakayama rule. To this end, we define admissible skew shapes for each F_i, each with a particular coefficient. In analogy with the original rule, admissible skew shapes play the role of border strips, and the coefficients of admissible skew shapes play the role of heights of border strips.

Definition 13. Given a set of partitions F, let Γ_F be the set of all skew shapes γ that admit a border strip tableau $T \in \mathcal{B}(\gamma, \rho)$ for some $\rho \in F$. For $\gamma \in \Gamma_F$, we define the coefficient

$$\alpha_F(\gamma) = \sum_{\rho \in F} \sum_{T \in \mathcal{B}(\gamma,\rho)} \mathrm{ht}(T).$$

In preparation for Section 6, we exemplify this definition with the cycle formats of active edge gadgets. For $F = \{(2^2), (4)\}$, we observe that the set Γ_F consists of all skew shapes that can be covered with two disjoint dominos. Note that there are shapes $\gamma \in \Gamma_F$ with $\alpha_F(\gamma) = 0$: For example, the vertical 4-box line admits a border strip tableau for (2^2) and one for (4), shown below. These tableaux have opposite height signs and thus cancel.



$$\mathrm{ht}(T) = 1 \qquad \mathrm{ht}(T') = -1 \qquad \Rightarrow \qquad \alpha_F(\gamma) = 0$$

This outcome is expected: In Section 6, we show that 4-box shapes γ with $\alpha_F(\gamma) \neq 0$ serve as a resource for establishing hardness of immanants. If we can peel many such shapes from a partition λ, then we can reduce a large permanent to the λ-immanant. Thus, if the vertical 4-box line γ satisfied $\alpha_F(\gamma) \neq 0$, our reductions would allow us to establish hardness of the determinant.

With all relevant notions introduced, we can now turn to our generalization of the Murnaghan-Nakayama rule. The relatively straightforward proof is given in the full version.

Lemma 14. *Let λ be a partition. Given a partition product $F_1 \times \ldots \times F_t$, abbreviate $\Gamma_i = \Gamma_{F_i}$ and $\alpha_i = \alpha_{F_i}$ for $i \in [t]$. Writing $\tilde{\Gamma} = (\Gamma_1, \ldots, \Gamma_t)$, we have*

$$\chi_\lambda(F_1 \times \ldots \times F_t) = \sum_{\substack{S \in \mathcal{S}(\lambda, \tilde{\Gamma}) \\ \text{with shapes } \gamma_1 \ldots \gamma_t}} \prod_{i=1}^{t} \alpha_i(\gamma_i). \tag{3}$$

4 STAIRCASES VERSUS TETROMINOS

We associate various quantities with partitions $\lambda \vdash n$ to measure to what extent the λ-immanant lends itself to a reduction from counting matchings. Then we investigate the interplay of these quantities with the number $b(\lambda)$ of boxes to the right of the first column of λ.

The diagrams □□ and ⊟ of (2^1) and (1^2) will be denoted as *domino*s. The domino number $d(\lambda)$ is the maximum number of dominos that can be peeled successively from λ. As we show below, the result of peeling these dominos from λ is a partition of the form $\mu = (k, \ldots, 1)$ for some $k \in \mathbb{N}$. Such partitions and their associated shapes are called *staircases*, and we define $z(\lambda) = k + \ldots + 1$ as the staircase size and $w(\lambda) = k$ as the staircase width of λ. Note that $2d(\lambda) + z(\lambda) = n$ and that $z(\lambda)$ can be zero; this happens when λ can be covered fully by dominos.

Fact 15. *The shape μ obtained by peeling $d(\lambda)$ dominos from λ is the unique staircase on $z(\lambda)$ boxes.*

Proof. By maximality of $d(\lambda)$, no domino can be peeled from μ, so all gaps between consecutive rows in μ are 1. This requires μ to be a staircase. It has $n - 2d(\lambda) = z(\lambda)$ boxes, and the number of boxes uniquely determines a staircase. □

We define $s(\lambda)$ to be the maximum number of *non-vanishing tetrominos* (depicted on page 5 and Figure 6) that can be peeled successively from λ. As we establish in Sections 5 and 6, any partition λ with large $w(\lambda)$ or $s(\lambda)$ induces an immanant to which the hard problem of counting k-matchings can be reduced. In the full version of the paper, we show that at least one of these reductions can be used, since at least one of $w(\lambda)$ or $s(\lambda)$ is large if $b(\lambda)$ is large.

Lemma 16. *For any partition λ, at least one of $s(\lambda) \geq b(\lambda)/8$ or $w(\lambda) \geq \sqrt{b(\lambda)} - 1$ holds.*

5 USING A STAIRCASE

We show how to count k-matchings in n-vertex graphs H with access to the λ-immanant for a partition λ with large staircase. For the parameterized reduction in the case of $k \ll n$, we also require a large domino number $d(\lambda)$. To simplify the presentation, we require a staircase width of $w(\lambda) \in \Omega(k)$, and thus, staircase size $z(\lambda) \in \Omega(k^2)$. At the end of the section, we sketch how the reductions can be modified to require only staircase size $z(\lambda) \in \Omega(k)$. The complete argument is contained in the full version.

Our reduction relies on the intermediate problem of counting cycle covers with a particular format (outlined in Section 5.1), which we then reduce to the λ-immanant using a graph construction that has a favorable interplay with staircase characters (in Section 5.2).

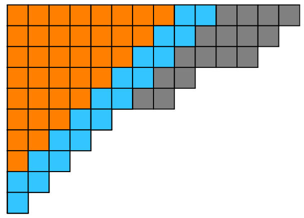


Fig. 3. A partition λ with $d(\lambda) = 8$ dominos, staircase width $w(\lambda) = 10$ and size $z(\lambda) = 55$. A partition $\rho^* = (2^8, 19, 1^{36})$ is shown as a border strip tableau; different dominos and singletons are not distinguished in this figure.

5.1 Main Construction

To reduce from counting k-matchings in a graph H, we use a particular partition ρ^* derived from λ, also depicted in Figure 3:

$$\rho^* = (2^{d(\lambda)}, 2k + 1, 1^{z(\lambda)-2k-1}).$$

In other words, this partition is obtained as follows: First peel all $d(\lambda)$ dominos from λ to expose the staircase μ. Then peel a border strip of length $2k + 1$ from μ. Finally, peel the remaining $z(\lambda) - 2k - 1$ boxes from μ as singletons. The k edges of the matching will be "accommodated" in the border strip of length $2k + 1$: Each edge requires two boxes, and an additional box is required to ensure an odd length of the border strip.

We describe two related ways of constructing a digraph G whose cycle covers of format ρ^* correspond to k-matchings in H. Given an edge-weighted digraph G, we write $\#\mathrm{CC}(G, \rho)$ for the immanant $\mathrm{imm}_f(G)$ associated with the class function $f : S_n \to \{0, 1\}$ that tests whether π has format ρ. That is,

$$\#\mathrm{CC}(G, \rho) = \sum_{\substack{\text{cycle cover } C \text{ of} \\ \text{format } \rho \text{ in } G}} \prod_{e \in C} w(e).$$

We write $[x^s]\,p$ for the coefficient of x^s in a polynomial $p \in \mathbb{Q}[x]$. We also say that a vertex-set T in a graph G is an *odd-cycle transversal* if $G - T$ is bipartite up to self-loops. (Note that self-loops are allowed in $G - T$ according to our definition, even though they are odd-length cycles.)

Lemma 17. *Let H be a bipartite n-vertex graph, for even n, and let λ be a partition of an integer $n' \geq n$.*

(1) *If $w(\lambda) \geq n/2 + 1$, let $\rho^* = (2^{d(\lambda)}, n + 1, 1^{z(\lambda)-n-1})$. We can construct a digraph G with*

$$\#\mathrm{CC}(G, \rho^*) = (n/2)! \cdot \#\mathrm{PerfMatch}(H)$$

in polynomial time. Every cycle cover in G contains at least $d(\lambda)$ digons and exactly $z(\lambda) - n - 1$ self-loops.

(2) *For any $k \in \mathbb{N}$ with $d(\lambda) \geq n - 2k$: If $w(\lambda) \geq k/2 + 1$ and $z(\lambda) \geq 4k + 1$, let $\rho^* = (2^{d(\lambda)}, 2k + 1, 1^{z(\lambda)-2k-1})$. We can construct a digraph G (whose edge-weights may be constant multiples of an indeterminate x) in polynomial time such that*

$$[x^{2k}]\,\#\mathrm{CC}(G, \rho^*) = k! \cdot \#\mathrm{Match}(H, k).$$

Every cycle cover whose weight is a constant multiple of x^{2k} contains at least $d(\lambda)$ digons and exactly $z(\lambda) - 2k - 1$ self-loops.

Furthermore, in both cases, G admits an odd-cycle transversal consisting of a single vertex.

Proof. Let $V(H) = L \cup R$. For the **first part** of the lemma, the graph G is defined as follows:

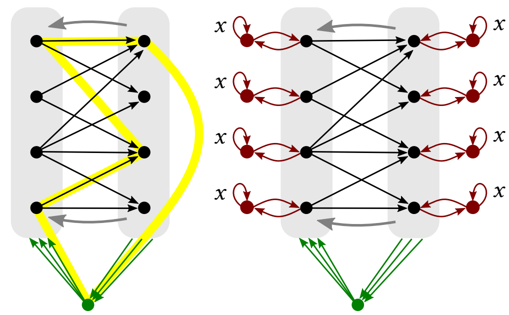


Fig. 4. The graphs from Lemma 17. To reduce clutter, edges from R to L are only hinted, and only one transit vertex is shown. In the left graph, a cycle corresponding to a 2-matching is displayed in yellow color.

(1) Direct all edges in H from L to R and add all edges $R \times L$.
(2) Add a *transit vertex* t and all edges in $R \times \{t\}$ and $\{t\} \times L$.
(3) Add $d(\lambda)$ disjoint *padding digons* and $z(\lambda) - n - 1$ *padding vertices* with self-loops.

Any ρ^*-cycle cover of G uses all padding digons and loops. The remaining cycle of odd length $n+1$ must use the transit vertex t, as it would otherwise be contained in a bipartite graph. Hence, any ρ^*-cycle cover C in G induces a perfect matching in H when restricted to edges from L to R: Deleting the transit vertex results in a cover of $V(H)$ with paths of odd length ≥ 1 that start in L, and deleting the edges from R to L then leaves us with a matching. Conversely, every perfect matching in H induces exactly $(n/2)!$ cycle covers of format ρ^* in G, as there are $(n/2)!$ ways of choosing an ordering of the edges and the transit vertex within the odd-length cycle.[3] This concludes the first part of the lemma.

For the **second part**, we construct G by performing steps 1 and 2 from above, with the following additional steps:

3. For each vertex $v \in V(H)$, add a *switch vertex* s_v with a self-loop of weight x, and a *switch digon* between v and s_v.
4. Add $d(\lambda) - (n - 2k)$ *padding digons* and $z(\lambda) - (2k+1) - 2k$ *padding vertices* with self-loops.

Any cycle cover C uses all padding elements. The weight of C is x^{2k} iff it includes exactly $2k$ self-loops at switch vertices; it then includes $n - 2k$ switch digons touching the remaining switch vertices, so it contains at least $d(\lambda)$ digons and $z(\lambda) - (2k+1)$ self-loops. As before, if C has format ρ^*, then the remaining odd-length cycle induces a k-matching in H when restricted to edges from L to R, and any k-matching in H can be extended to $k!$ such cycle covers. □

In the next subsection, we use the odd-cycle transversal of G (consisting of the transit vertex) and special properties of staircase characters to reduce $\#\mathrm{CC}(G, \rho^*)$ for the partitions ρ^* described above to $\mathrm{imm}_\lambda(G)$.

5.2 Staircase Characters

Given a partition λ with staircase μ, consider any partition $\rho = (2^{d(\lambda)}, \rho')$ obtained by peeling the maximum number of dominos from λ, followed by peeling some other partition ρ' of the staircase μ thusly exposed. We show how to relate $\chi_\lambda(\rho)$ to $\chi_\mu(\rho')$ by means of *domino tilings* of λ/μ, which are border strip tableaux that only contain the dominos ⊟ and ◫.

[3] Each such ordering corresponds to a pair (j, σ) for $j \in [n/2]$ and a cyclic permutation σ of $[n/2]$. The index j indicates that the transit vertex t_i is visited immediately after the j-th edge, and σ describes the order of the edges along the cycle. There are $n/2 \cdot (n/2-1)! = n/2!$ such pairs (j, σ).

Definition 18. A *domino tiling* of a skew shape λ/μ with t boxes is a border strip tableau T of λ/μ with format $(2^{t/2})$. The *parity* of a domino tiling T is the parity of the number of $\boxminus$ in T.

Note that the parity of a domino tiling T is even/odd iff the height sign $\mathrm{ht}(T)$ is positive/negative. Curiously, the parity does not depend on T.

Lemma 19. *All domino tilings of fixed λ/μ have the same parity.*

Proof. Paint the rows of λ/μ black and white in an alternating way. In any domino tiling T, every vertical (horizontal) domino contains an odd (even) number of white boxes. Hence, the number of vertical dominos in T agrees in parity with the number of white boxes in λ/μ, which does not depend on T. □

This gives the desired connection between $\chi_\lambda(\rho)$ and $\chi_\mu(\rho')$.

Lemma 20. *Let λ be a partition with staircase μ, and let $\rho = (2^{d(\lambda)}, \rho')$ for a partition ρ'. Then we have $\chi_\lambda(\rho) \neq 0$ iff $\chi_\mu(\rho') \neq 0$.*

Proof. As any way of peeling $d(\lambda)$ dominos from λ results in μ, the Murnaghan-Nakayama rule (Theorem 10) shows that $\chi_\lambda(\rho) = c_{\lambda/\mu} \cdot \chi_\mu(\rho')$ with

$$c_{\lambda/\mu} = \sum_{\substack{\text{domino tiling} \\ T \text{ of } \lambda/\mu}} \mathrm{ht}(T), \tag{4}$$

By Lemma 19, each term in (4) has the same sign. Since λ/μ is obtained by peeling $d(\lambda)$ dominos from λ, there is at least one domino tiling, and hence there is at least one term in the sum. It follows that $c_{\lambda/\mu} > 0$, thus proving the lemma. □

The last missing piece is to recall Remark 11: The staircase character $\chi_\mu(\rho')$ vanishes whenever ρ' contains an even part. This has consequences for the cycle covers in the graphs constructed in the last subsection. Namely, provided that the right number of self-loops is enforced, such a cycle cover is counted by $\mathrm{imm}_\lambda(G)$ iff its format is ρ^*.

Lemma 21. *Let $\lambda \vdash n'$ be a partition and let $\rho^* = (2^{d(\lambda)}, s, 1^{z(\lambda)-s})$ for $s \in \mathbb{N}$. Let G be an n'-vertex digraph with a single-vertex odd-cycle transversal. For $t \in \mathbb{N}$, if every cycle cover of G whose weight is a constant multiple of x^t contains at least $d(\lambda)$ digons and exactly $z(\lambda) - s$ self-loops, then*

$$[x^t]\,\mathrm{imm}_\lambda(G) = \chi_\lambda(\rho^*) \cdot [x^t]\,\#\mathrm{CC}(G, \rho^*) \tag{5}$$

with $\chi_\lambda(\rho^*) \neq 0$.

Proof. We first show that any cycle cover C in G with $\chi_\lambda(C) \neq 0$ that contains at least $d(\lambda)$ digons and exactly $z(\lambda) - s$ self-loops has format ρ^*. This implies (5). To this end, let C be a cycle cover in G with format $\beta = (2^{d(\lambda)}, \rho')$ and $\chi_\lambda(\beta) \neq 0$. By Lemma 20, we have $\chi_\mu(\rho') \neq 0$, where μ is the staircase of λ. Since G has an odd-cycle transversal of size 1, there is at most one non-singleton odd part in ρ', as every non-singleton odd cycle uses exactly one transversal vertex. Since ρ' contains $z(\lambda) - s$ singletons, the non-singleton odd part contains s boxes, so we have $\rho' = (s, 1^{z(\lambda)-s})$.

Next, to show $\chi_\lambda(\rho^*) \neq 0$, it suffices by Lemma 20 to show $\chi_\mu(\rho') \neq 0$ for $\rho' = (\theta, 1^{z(\lambda)-s})$. Each way of peeling a border-strip of length s and the $z(\lambda) - s$ singletons from ρ' incurs positive height sign. It follows that all border strip tableaux of format ρ' in μ contribute to $\chi_\mu(\rho')$ with the same sign. □

Note that we can consider G to be unweighted and invoke the lemma with $t = 0$.

5.3 Reductions

We combine the results from the previous sections into reductions from counting matchings to evaluating immanants for partitions with large staircases.

Lemma 22. *The following can be achieved in polynomial time, given a bipartite n-vertex graph H:*

(1) *Given a partition λ with $w(\lambda) \geq 2\sqrt{n}$, compute a digraph G and a number $c \in \mathbb{Q}$ such that* $\#\mathrm{PerfMatch}(H) = c \cdot \mathrm{imm}_\lambda(G)$.
(2) *Given $k \in \mathbb{N}$ and a partition λ with $w(\lambda) \geq 4\sqrt{k}$ and $d(\lambda) \geq n - 2k$, compute a digraph G and a number $c \in \mathbb{Q}$ such that* $\#\mathrm{Match}(H, k) = c \cdot [x^{2k}]\, \mathrm{imm}_\lambda(G)$.

Proof sketch. For ease of exposition, we prove a weaker version of this lemma, where we assume stronger lower bounds on $w(\lambda)$. This is non-essential for the hardness results for #P or VNP, and it matters only for the lower bounds under #ETH. The proof of the statement with the bounds stated above is given in the full version.

For the first part of the lemma, we make the stronger assumption that $w(\lambda) \geq n + 1$. We combine Lemmas 17 and 21 (invoked with $t = 0$) to construct a digraph G and a partition ρ^* with

$$\begin{aligned}\mathrm{imm}_\lambda(G) &= \chi_\lambda(\rho^*) \cdot \#\mathrm{CC}(G, \rho^*)\\ &= \chi_\lambda(\rho^*) \cdot (n/2)! \cdot \#\mathrm{PerfMatch}(H)\end{aligned}$$

such that $c = \chi_\lambda(\rho^*) \cdot (n/2)! \neq 0$. This value can be computed as $c = \mathrm{imm}_\lambda(F)$, where F is the graph obtained by invoking Lemma 17 on the $n/2$-edge matching $M_{n/2}$ with $\#\mathrm{PerfMatch}(M_{n/2}) = 1$.

For the second part, we assume that $w(\lambda) \geq k + 1$; then all conditions for the second part of Lemma 17 are fulfilled. We can thus construct a graph G with

$$\begin{aligned}[x^{2k}]\, \mathrm{imm}_\lambda(G) &= \chi_\lambda(\rho^*) \cdot [x^{2k}]\, \#\mathrm{CC}(G, \rho^*)\\ &= \chi_\lambda(\rho^*) \cdot k! \cdot \#\mathrm{Match}(H, k).\end{aligned}$$

This proves the lemma. □

Let us remark how to prove the full statement of Lemma 22. In the reduction shown in this section, only one cycle of length $2k + 1$ is peeled from the staircase μ, so only $\Theta(w(\lambda))$ boxes are available for accommodating a matching. We could instead peel cycles of length $2k + 1, 2k - 3, \ldots$ to use most of the staircase μ. This requires several transit vertices and a more complicated character analysis, but ultimately this allows us to reduce an instance (H, k) for #Match to imm_λ for a partition λ with staircase *size* $z(\lambda) = O(k)$ rather than *width*. Please see the full version for this proof.

6 USING NON-VANISHING TETROMINOS

We show how to count k-matchings with access to λ-immanants for partitions λ with large non-vanishing tetromino number $s(\lambda)$: In Section 6.1, we use edge gadgets to construct particular immanants $\mathrm{imm}_\lambda(G)$ that count matchings up to a multiplicative constant factor. We then show, in Section 6.2, that the factor arising in the above construction is non-zero. To this end, we prove that the character χ_λ does not vanish on a particular partition product. Then we combine these insights in Section 6.3 to obtain a reduction from counting matchings.

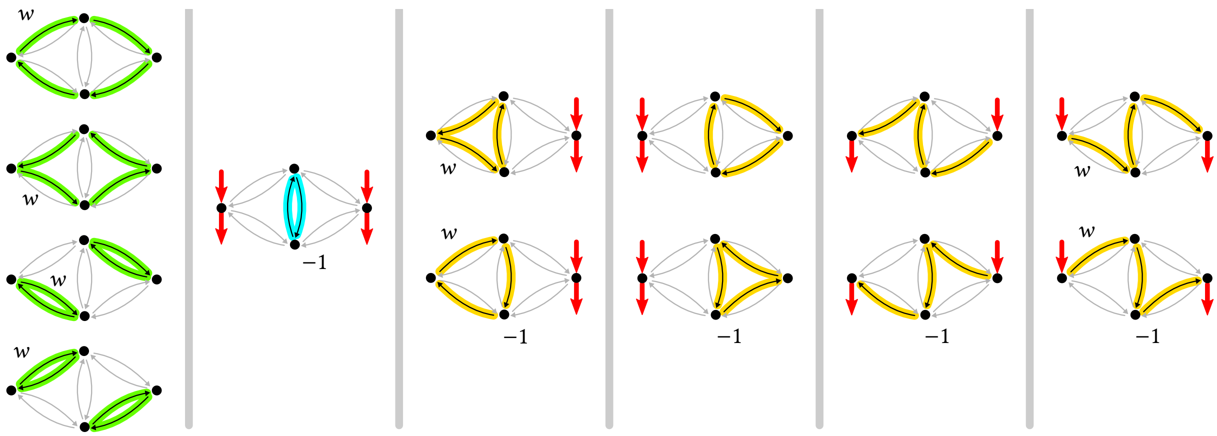


Fig. 5. Each column lists the edge-sets C_i that can arise for fixed combinations of endpoint in/out-degrees at gadget Q_i in $E(G) \setminus E(Q_i)$. The first column lists *active* states, the second column shows the *passive* state. All other states come with annihilating partners and cancel out, leading to a proof of Lemma 23.

6.1 Main Construction

Consider the edge gadget Q depicted on page 6. Intuitively speaking, this graph fragment ensures that unwanted cycle covers in G annihilate in $\mathrm{imm}_\lambda(G)$. In the remaining *consistent* cycle covers C, the edges within each edge gadget Q cover either *both* endpoints (in an active state, as listed in Figure 5) or *none* of them (in a passive state). In particular, this allows us to interpret an active gadget Q as a matching edge between its endpoints, since the active state prevents any edge outside of Q from being incident with the endpoints of Q. A similar approach was taken by the author together with Bläser [3] to establish hardness of the so-called *cover polynomial*.

Lemma 23. *Let G be a directed graph containing copies $Q_1, \ldots, Q_t$ of the edge-gadget Q, with endpoints $e_i = \{u_i, v_i\}$ for $i \in [t]$, such that distinct edge-gadgets intersect only at endpoints. Let $C^*(G)$ denote the set of* consistent *cycle covers C in G: In such cycle covers, the restriction $C \cap E(Q_i)$ for $i \in [t]$ is an active or passive state, as depicted in Figure 5. Then we have*

$$\mathrm{imm}^*_\lambda(G) := \sum_{C \in C^*(G)} \chi_\lambda(C) \prod_{e \in C} w(e) \;=\; \mathrm{imm}_\lambda(G).$$

The proof is given in the full version, but Figure 5 summarizes the main idea: Inconsistent cycle covers annihilate due to the special form of the edge-gadget.

Using Lemma 23, we show how to transform instances (H, k) for #Match into digraphs G such that $\mathrm{imm}_\lambda(G)$ equals $\#\mathrm{Match}(H, k)$ up to a multiplicative constant. This constant needs some attention: To define it, consider the partition products $\{(2^2), (4)\}^{\times s}$ induced by the cycle formats of active states; we pad the partitions in this product to partitions of n' with $d - 2s$ dominos and $z(\lambda)$ singletons. Formally, for fixed $\lambda \vdash n'$ and $s \leq s(\lambda)$, define

$$\theta_s := \{(2^2), (4)\}^{\times s} \times \{(2^{d(\lambda)-2s}, 1^{z(\lambda)})\}. \tag{6}$$

In the next subsection, we then establish the crucial fact that $\chi_\lambda(\theta_s) \neq 0$ for relevant choices of s.

Lemma 24. *Let H be a graph with n vertices and m edges and let $\lambda \vdash n'$ be a partition.*

(1) *If λ has skew tetromino number $s(\lambda) \geq n/2$ and $m - n/2$ additional dominos, i.e., $d(\lambda) \geq n/2 + m$, then we can construct an n'-vertex graph G in polynomial time such that*

$$\mathrm{imm}_\lambda(G) = \underbrace{(-1)^{m-n/2} \cdot 2^{n/2} \cdot \chi_\lambda(\theta_{n/2})}_{=:c_1} \cdot \#\mathrm{PerfMatch}(H). \tag{7}$$

(2) *For any $k \leq \frac{s(\lambda)}{3}$ such that $d(\lambda) \geq m + n + 2kn - 5k$: We can construct an n'-vertex graph G in polynomial time such that*

$$\mathrm{imm}_\lambda(G) = \underbrace{(-1)^{m+2kn-3k} \cdot (2k)! \cdot 2^{3k} \cdot \chi_\lambda(\theta_{3k})}_{=:c_2} \cdot \#\mathrm{Match}(H, k).$$

Proof. For the **first part**, we define G as follows:

(1) Replace each edge $uv \in E(H)$ with a fresh copy Q_{uv} of the edge gadget Q. Identify u and v with the endpoints of Q.
(2) Add $d' = d(\lambda) - (n/2 + m)$ *padding digons* and $z(\lambda)$ isolated *padding vertices* with self-loops.

By Lemma 23 we have $\mathrm{imm}_\lambda(G) = \mathrm{imm}^*_\lambda(G)$, where $\mathrm{imm}^*_\lambda(G)$ sums over the set $\mathcal{C}^*$ of consistent cycle covers. Any cycle cover $C \in \mathcal{C}^*$ includes all padding elements. Apart from padding elements, C consists of the active states of some gadgets and the passive states of the remaining gadgets; let $M(C) \subseteq E(H)$ denote the set of pairs uv such that Q_{uv} is active in C. Since active gadget states must be vertex-disjoint, and all vertices of G must be covered by cycles in C, the set $M(C)$ induces a perfect matching $M(C) \in \mathcal{M}_{n/2}(H)$ in H. Conversely, given a perfect matching M of H, let $\mathcal{C}^*_M \subseteq \mathcal{C}^*$ denote the set of consistent cycle covers with $M(C) = M$. By grouping the terms in $\mathrm{imm}^*_\lambda(G)$, we obtain

$$\mathrm{imm}^*_\lambda(G) = \sum_{M \in \mathcal{M}_{n/2}(H)} \underbrace{\sum_{C \in \mathcal{C}^*_M} \chi_\lambda(C) \cdot w(C)}_{=:a(M)}. \tag{8}$$

To calculate $a(M)$, we investigate the set $\mathcal{C}^*_M$: Each cycle cover $C \in \mathcal{C}^*_M$ is obtained by (i) choosing an active state for each of the $n/2$ gadgets Q_{uv} with $uv \in M$, each inducing the weight $w(uv)$, then (ii) adding the passive state (of weight -1) at the remaining gadgets, for a total weight of $(-1)^{m-n/2}$, and finally (iii) adding all padding elements, all of weight 1.

The total weight is thus $(-1)^{m-n/2} w(M)$. As the choices in the above process can only be made at active states, the formats of cycle covers in $\mathcal{C}^*_M$ are given by the partition product

$$\{(2^2), (2^2), (4), (4)\}^{\times n/2} \times \{(2^{d(\lambda)-n}, 1^{z(\lambda)})\}.$$

As a multiset of partitions, this product amounts to $2^{n/2}$ copies of $\theta_{n/2}$: After choosing one of the formats $\{(2^2), (4)\}$ for each of $n/2$ entries, we can choose the first or second copy of this format. This shows that $a(M) = (-1)^{m-n/2} \cdot 2^{n/2} \cdot \chi_\lambda(\theta_{n/2}) \cdot w(M)$ and thus proves (7).

For the **second part**, we construct the graph G in a similar way, but we need to add some additional structures to account for the fact that most vertices are unmatched in a k-matching for $k \ll n$.

(1) Replace all edges of H by edge gadgets.
(2) For each vertex $v \in V(H)$, add a *switch vertex* s_v and connect it to v with a *switch digon*.
(3) Add $2k$ *receptor vertices*. Add a receptor edge between each pair of receptor and switch vertex, then replace these edges by edge gadgets.

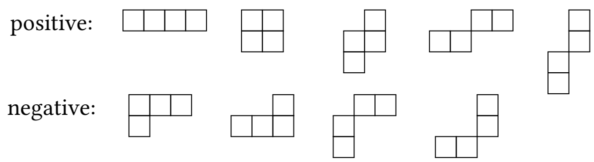


Fig. 6. The non-vanishing tetrominos are the skew shapes γ with $\alpha_F(\gamma) \neq 0$ for $F = \{(2^2), (4)\}$, here grouped by sign.

(4) Add $d' = d(\lambda) - (m + n + 2kn - 5k)$ isolated digons and $z(\lambda)$ isolated vertices.

Any cycle cover $C \in \mathcal{C}^*$ then consists of the following cycles:

- Each receptor vertex must be covered by an active edge gadget. Then the other endpoint of that gadget is the switch vertex of some vertex in H. The remaining $2k \cdot (n-1)$ edge gadgets incident with receptor vertices are passive. There are $(2k)!$ ways of matching the $2k$ receptor vertices to $2k$ fixed switch vertices with active gadgets.
- The $n - 2k$ switch vertices not touched by active gadgets from receptor vertices must be covered by switch digons.
- By the previous item, $2k$ vertices in H are left to be covered by k active edge gadgets that represent edges in H. As active edge gadgets are vertex-disjoint, they induce a k-matching $M(C)$ in H, and they contribute weight $\prod_{uv \in M(C)} w(uv)$.
- Overall, there are $m - k + 2k \cdot (n-1) = m + 2kn - 3k$ passive edge gadgets in C, each contributing weight -1. With padding digons and loops in G, there are $d(\lambda) - 6k$ digons and $z(\lambda)$ loops in C.

The third item describes how C induces a k-matching $M(C) \in \mathcal{M}_k(H)$. Conversely, we can observe (as in the first case) that any k-matching M of H induces consistent cycle covers with a total contribution of $(-1)^{m+2kn-3k} \cdot (2k)! \cdot 2^{3k} \cdot w(M) \cdot \chi_\lambda(\theta_{3k})$. Note that the factor $(2k)!$ stems from the different ways receptor vertices can match to switch vertices. ◻

6.2 Analyzing the Character Values

In this section, we analyze $\chi_\lambda(\theta_s)$ for the partition product θ_s defined in (6). In the following, recall that a *domino tiling* of a $2t$-box skew shape γ is a border strip tableau of format (2^t). We call it *even/odd* if its number of vertical $\boxminus$ pieces is even/odd. By Lemma 19, all domino tilings of γ have the same parity. Also recall Definition 13 for the skew shapes Γ_F and the coefficients α_F derived from a set F of partitions.

Fact 25. *For $F = \{(2^2), (4)\}$, we have $\alpha_F(\gamma) \neq 0$ iff γ is a non-vanishing tetromino, that is, one of the shapes listed in Figure 6. Furthermore, for any non-vanishing tetromino, the sign of $\alpha_F(\gamma)$ is given by the parity of its domino tilings, see Figure 6:*

$$\alpha_F(\gamma) = \begin{cases} +2 & \gamma \text{ has even domino tilings,} \\ -2 & \gamma \text{ has odd domino tilings.} \end{cases}$$

We can now prove that the relevant characters in the tetromino-based reduction do not vanish.

Lemma 26. *For any $\lambda \vdash n'$ and $s \leq s(\lambda)$, we have $\chi_\lambda(\theta_s) \neq 0$.*

Proof. By Fact 25, we have $\gamma \in \Gamma_F$ and $\alpha_F(\gamma) \neq 0$ for a 4-box skew shape γ iff γ is a non-vanishing tetromino. Let $D = \{(2)\}$ and $S = \{(1)\}$; then $\Gamma_D = \{\square\square, \boxminus\}$ and $\Gamma_S = \{\square\}$. Define the set of skew shape tableaux

$$\mathcal{S} = \mathcal{S}(\lambda, \underbrace{\Gamma_F, \dots, \Gamma_F}_{s \text{ times}}, \underbrace{\Gamma_D, \dots, \Gamma_D}_{d(\lambda)-2s \text{ times}}, \underbrace{\Gamma_S, \dots, \Gamma_S}_{z(\lambda) \text{ times}})$$

and let $t = d(\lambda) - s + z(\lambda)$ be the number of sets of skew shapes in the above list. For $i \in [t]$, let $\alpha_i \in \{\alpha_F, \alpha_D, \alpha_S\}$ be the coefficient function for the i-th set in the list. By Lemma 14, we have

$$\chi_\lambda(\theta_s) = \sum_{\substack{S \in \mathcal{S} \text{ with} \\ \text{shapes } \gamma_1 \dots \gamma_t}} \prod_{i=1}^{t} \alpha_i(\gamma_i). \tag{9}$$

It follows that every tableau $S \in \mathcal{S}$ with non-zero weight in the above sum peels s non-vanishing tetrominos from λ, followed by $d(\lambda) - 2s$ dominos, and $z(\lambda)$ singleton boxes. The tetrominos and dominos tile λ/μ, where μ is the staircase of λ.

Since $s \leq s(\lambda)$, at least one tableau $S \in \mathcal{S}$ exists, and we can prove the lemma by showing that all tableaux are counted with the same sign in (9). Towards this, note that any skew shape tableau $S \in \mathcal{S}$ can be turned into a border strip tableau $B(S)$ of λ that peels $d(\lambda)$ dominos and $z(\lambda)$ singleton boxes from λ: Simply tile each tetromino γ in S arbitrarily with two dominos. By Fact 25, we know that $\alpha_F(\gamma)$ is positive/negative iff the tiling of γ is even/odd, so $\prod_{i=1}^{t} \alpha_i(\gamma_i)$ is positive/negative iff the domino tiling of λ/μ induced by $B(S)$ is even/odd. (Singleton boxes can be ignored, as they contribute the factor +1.) But by Lemma 19, all domino tilings of λ/μ have the same parity. Therefore, all terms in (9) have the same sign. □

Corollary 27. *The coefficients c_1 and c_2 defined in Lemma 24 are both non-zero.*

6.3 Reductions

As in Section 5.3, we collect the previous arguments to obtain a reduction from counting matchings to immanants for partitions with large non-vanishing tetromino number.

Lemma 28. *The following can be achieved in polynomial time and with polynomial-sized arithmetic circuits:*

(1) *Given an n-vertex graph H of maximum degree 3 and a partition λ with $s(\lambda) \geq n/2$ and $d(\lambda) \geq 2n$, compute a digraph G and a number $c \in \mathbb{Q}$ such that $\#\mathrm{PerfMatch}(H) = c \cdot \mathrm{imm}_\lambda(G)$.*

(2) *Given an n-vertex graph H and $k \in \mathbb{N}$, and a partition λ with $s(\lambda) \geq 3k$ and $d(\lambda) \geq n^2 + n + 2kn - 5k$, compute a digraph G and a number $c \in \mathbb{Q}$ such that $\#\mathrm{Match}(H, k) = c \cdot \mathrm{imm}_\lambda(G)$.*

Proof. For the first part, let H be a graph with n vertices and maximum degree 3, so that $m \leq \frac{3}{2}n$. Note that $d(\lambda) \geq n + m$ by assumption. We construct a graph G via Lemma 24 with $\mathrm{imm}_\lambda(G) = c_1 \cdot \#\mathrm{PerfMatch}(H)$. For the second part, we use Lemma 24 to construct a graph G such that $\mathrm{imm}_\lambda(G) = c_2 \cdot \#\mathrm{Match}(H, k)$. We have $c_1, c_2 \neq 0$ by Corollary 27 and can compute these values as in the proof of Lemma 22. □

7 COMPLETING THE PROOFS

Let Λ be a family of partitions with unbounded $b(\Lambda)$. We compose the constructions from the preceding sections to an overall hardness proof for $\mathrm{Imm}(\Lambda)$. This requires us to find sequences of partitions within Λ that are dense enough and supply sufficiently many boxes to the right of the first column. In the sub-polynomial growth regime, we also need to ensure sufficiently many dominos; this can be achieved by having a large number of boxes in the first column.

Definition 29. Given a polynomial-time computable function $g : \mathbb{N} \to \mathbb{N}$, a family of partitions Λ *supports growth* g if there is a sequence $\Lambda' = (\lambda^{(1)}, \lambda^{(2)}, \ldots)$ in Λ such that $\lambda^{(n)}$ has $\Theta(n)$ boxes and satisfies $b(\lambda^{(n)}) \geq g(n)$. We also say that Λ supports growth g *via* Λ'. We say that Λ *computationally* supports growth g if $\lambda^{(n)}$ can be computed in polynomial time from n.

To prove the main theorems, we distinguish whether Λ supports polynomial growth (for a reduction from #PerfMatch) or only sub-polynomial growth (for a reduction from #Match$^{(g)}$ for any growth g supported by Λ). Recall that, by known algorithms [8, 23], we have $\mathrm{Imm}(\Lambda) \in \mathsf{FP}$ and $\mathrm{Imm}(\Lambda) \in \mathsf{VP}$ for any family Λ with $b(\Lambda) < \infty$.

Proof of Theorem 1. We reduce from #PerfMatch to $\mathrm{Imm}(\Lambda)$: Let H be an n-vertex bipartite graph for which we want to compute #PerfMatch(H). Let $\alpha > 0$ be maximal such that Λ supports growth $\Omega(n^{\alpha})$. We find a partition $\lambda \in \Lambda$ with $t = \Theta(n^{1/\alpha})$ boxes such that $b(\lambda) \geq 20n$. Then Lemma 16 guarantees that $s(\lambda) \geq 3.5n$ or $w(\lambda) \geq \sqrt{20n} - 1$ holds. In either case, using the first cases of Lemmas 22 and 28, we compute a t-vertex graph G and a number $c \in \mathbb{Q}$ with $\#\mathrm{PerfMatch}(H) = c \cdot \mathrm{imm}_{\lambda}(G)$. Overall, this yields polynomial-time and c-reductions from #PerfMatch to $\mathrm{Imm}(\Lambda)$, showing #P-hardness and VNP-completeness of the latter. The lower bound under #ETH follows, since $t = \Theta(n^{1/\alpha})$. ◻

If Λ supports only sub-polynomial growth, the proof proceeds similarly. In this case, we can find a sequence of partitions in which most rows have width 1. This allows us to peel a large amount of dominos from the left-most column.

Proof of Theorem 2. If Λ supports polynomial growth, we use Theorem 1. Otherwise, let $g \in \omega(1)$ be a growth supported by Λ. We reduce from #Match$^{(h)}$ with $h(n) = \sqrt{g(n)}/24$: Let (H, k) be an instance for #Match$^{(h)}$ with an n-vertex graph H and $k \leq h(n)$. Using the growth condition on Λ and $g \in O(n^{0.1})$, we find a partition $\lambda \in \Lambda$ with $b(\lambda) \geq 24k$ and at least $2n^2 + b(\lambda)$ boxes in the first column, which implies $d(\lambda) \geq n^2$. With the bound on $b(\lambda)$, Lemma 16 yields that $s(\lambda) \geq 3k$ or $w(\lambda) \geq \sqrt{24k} - 1$. In either case, using the second case of Lemmas 22 and 28, we compute a graph G on $\|\lambda\|$ vertices and a number $c \in \mathbb{Q}$ such that $\#\mathrm{Match}(H, k) = c \cdot [x^t]\,\mathrm{imm}_{\lambda}(G)$, where $t = 2k$ for the staircase-based reduction (Lemma 22) and $t = 0$ for the tetromino-based reduction (Lemma 28).

Note that the value $[x^t]\,\mathrm{imm}_{\lambda}(G)$ can be computed by means of polynomial interpolation from the values $\mathrm{imm}_{\lambda}(G_i)$ for $i \in \{0, \ldots, \|\lambda\|\}$, where G_i is the graph obtained from G by replacing each edge of weight x with an edge of weight i. Overall, we obtain a polynomial-time Turing reduction from #Match$^{(h)}$ to $\mathrm{Imm}(\Lambda)$, which implies by Lemma 4 that $\mathrm{Imm}(\Lambda) \notin \mathsf{FP}$ unless $\mathsf{FPT} = \#\mathsf{W}[1]$. An analogous statement holds in the algebraic setting: As polynomial interpolation amounts to solving a system of linear equations, which can be performed with polynomial-sized circuits, we obtain a parameterized c-reduction from the p-family #Match$^{(h)}$ to the p-family $\mathrm{Imm}(\Lambda')$. ◻

8 CONCLUSION AND FUTURE WORK

We established a dichotomy for the complexity of immanants, concluding a sequence of previously obtained partial results. We note that immanants are not the only way of interpolating between permanents and determinants. Other examples include the *cover polynomials* [5, 14] and the *fermionants* [2, 27], which are also sum-products over row-column permutations of a matrix. These families of matrix forms however do *not* exhibit gradual progressions from easy to hard cases, and they feature no non-trivial easy cases beside the determinant.

Our dichotomy for immanants also prompts several follow-up questions, some of which were partially answered by Miklós and Riener [29] since submission of this article.

Modular immanants. It is known that λ-immanants are tractable over $\mathbb{Z}_2$ for partitions λ with constantly many boxes outside of the first column *or row*. Are these the only tractable immanants over $\mathbb{Z}_2$? Which immanants are tractable over $\mathbb{Z}_p$ for odd primes p?

Planar graphs. The permanent is polynomial-time solvable for bi-adjacency matrices of planar bipartite graphs [36]. Can this be generalized to other immanants of bi-adjacency matrices of planar bipartite graphs? Which immanants remain hard on planar graphs?

Removing weights. Our proof establishes hardness for matrices with general entries from $\mathbb{Z}$, as the edge-gadget and interpolation steps introduce non-unit weights. It may however still be possible to establish hardness for 0-1 matrices, as known for the permanent.

ACKNOWLEDGMENTS

This work was supported by VILLUM Foundation grant 16582. I thank Christian Engels for introducing me to the immanants and Nathan Lindzey for introducing me to the representation theory of the symmetric group.

REFERENCES

[1] A. I. Barvinok. 1990. Computational complexity of immanants and representations of the full linear group. *Functional Analysis and Its Applications* 24 (1990), 144–145. https://doi.org/10.1007/BF01077707

[2] Andreas Björklund, Petteri Kaski, and Ryan Williams. 2019. Generalized Kakeya sets for polynomial evaluation and faster computation of fermionants. *Algorithmica* 81, 10 (2019), 4010–4028. https://doi.org/10.1007/s00453-018-0513-7

[3] Markus Bläser and Radu Curticapean. 2011. The Complexity of the Cover Polynomials for Planar Graphs of Bounded Degree. In *Mathematical Foundations of Computer Science 2011 - 36th International Symposium, MFCS 2011, Warsaw, Poland, August 22-26, 2011. Proceedings (Lecture Notes in Computer Science)*, Filip Murlak and Piotr Sankowski (Eds.), Vol. 6907. Springer, 96–107. https://doi.org/10.1007/978-3-642-22993-0_12

[4] Markus Bläser and Radu Curticapean. 2012. Weighted Counting of k-Matchings Is #W[1]-Hard. In *Parameterized and Exact Computation - 7th International Symposium, IPEC 2012, Ljubljana, Slovenia, September 12-14, 2012. Proceedings (Lecture Notes in Computer Science)*, Dimitrios M. Thilikos and Gerhard J. Woeginger (Eds.), Vol. 7535. Springer, 171–181. https://doi.org/10.1007/978-3-642-33293-7_17

[5] Markus Bläser, Holger Dell, and Mahmoud Fouz. 2012. Complexity and Approximability of the Cover Polynomial. *Comput. Complex.* 21, 3 (2012), 359–419. https://doi.org/10.1007/s00037-011-0018-0

[6] Markus Bläser and Christian Engels. 2019. Parameterized Valiant's Classes. In *14th International Symposium on Parameterized and Exact Computation, IPEC 2019, September 11-13, 2019, Munich, Germany (LIPIcs)*, Bart M. P. Jansen and Jan Arne Telle (Eds.), Vol. 148. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 3:1–3:14. https://doi.org/10.4230/LIPIcs.IPEC.2019.3

[7] Jean-Luc Brylinski and Ranee Brylinski. 2003. Complexity and Completeness of Immanants. *CoRR* cs.CC/0301024 (2003). http://arxiv.org/abs/cs/0301024

[8] Peter Bürgisser. 2000. *Completeness and Reduction in Algebraic Complexity Theory.* Algorithms and computation in mathematics, Vol. 7. Springer.

[9] Peter Bürgisser. 2000. The Computational Complexity of Immanants. *SIAM J. Comput.* 30, 3 (2000), 1023–1040. https://doi.org/10.1137/S0097539798367880

[10] Peter Bürgisser. 2000. The Computational Complexity to Evaluate Representations of General Linear Groups. *SIAM J. Comput.* 30, 3 (2000), 1010–1022. https://doi.org/10.1137/S0097539798367892

[11] Peter Bürgisser, Michael Clausen, and Mohammad Amin Shokrollahi. 1997. *Algebraic complexity theory.* Grundlehren der mathematischen Wissenschaften, Vol. 315. Springer.

[12] Jin-yi Cai, Xi Chen, and Dong Li. 2008. A quadratic lower bound for the permanent and determinant problem over any characteristic $\neq 2$. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, Cynthia Dwork (Ed.). ACM, 491–498. https://doi.org/10.1145/1374376.1374446

[13] Gordon G. Cash. 2003. Immanants and Immanantal Polynomials of Chemical Graphs. *J. Chem. Inf. Comput. Sci.* 43, 6 (2003), 1942–1946. https://doi.org/10.1021/ci0300238

[14] Fan R. K. Chung and Ronald L. Graham. 1995. On the Cover Polynomial of a Digraph. *J. Comb. Theory, Ser. B* 65, 2 (1995), 273–290. https://doi.org/10.1006/jctb.1995.1055

[15] Radu Curticapean. 2013. Counting Matchings of Size k Is #W[1]-Hard. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I (Lecture Notes in Computer Science)*, Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg (Eds.), Vol. 7965. Springer, 352–363. https://doi.org/10.1007/978-3-642-39206-1_30

[16] Radu Curticapean. 2015. Block Interpolation: A Framework for Tight Exponential-Time Counting Complexity. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I (Lecture Notes in Computer Science)*, Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann (Eds.), Vol. 9134. Springer, 380–392. https://doi.org/10.1007/978-3-662-47672-7_31

[17] Radu Curticapean. 2015. *The simple, little and slow things count: on parameterized counting complexity.* Ph.D. Dissertation. Saarland University. http://scidok.sulb.uni-saarland.de/volltexte/2015/6217/

[18] Radu Curticapean, Holger Dell, and Dániel Marx. 2017. Homomorphisms are a good basis for counting small subgraphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, Hamed Hatami, Pierre McKenzie, and Valerie King (Eds.). ACM, 210–223. https://doi.org/10.1145/3055399.3055502

[19] Radu Curticapean and Dániel Marx. 2014. Complexity of Counting Subgraphs: Only the Boundedness of the Vertex-Cover Number Counts. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*. IEEE Computer Society, 130–139. https://doi.org/10.1109/FOCS.2014.22

[20] Nicolas de Rugy-Altherre. 2013. Determinant versus Permanent: Salvation via Generalization?. In *The Nature of Computation. Logic, Algorithms, Applications - 9th Conference on Computability in Europe, CiE 2013, Milan, Italy, July 1-5, 2013. Proceedings (Lecture Notes in Computer Science)*, Paola Bonizzoni, Vasco Brattka, and Benedikt Löwe (Eds.), Vol. 7921. Springer, 87–96. https://doi.org/10.1007/978-3-642-39053-1_10

[21] Jörg Flum and Martin Grohe. 2004. The Parameterized Complexity of Counting Problems. *SIAM J. Comput.* 33, 4 (2004), 892–922. https://doi.org/10.1137/S0097539703427203

[22] W. Fulton, W.F.J. Harris, and J. Harris. 1991. *Representation Theory: A First Course.* Springer New York.

[23] W. Hartmann. 1985. On the complexity of immanants. *Linear and Multilinear Algebra* 18, 2 (1985), 127–140. https://doi.org/10.1080/03081088508817680 arXiv:https://doi.org/10.1080/03081088508817680

[24] D. E. Littlewood and A. R. Richardson. 1934. Group Characters and Algebra. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 233 (1934), 99–141. http://www.jstor.org/stable/91293

[25] M. Lorenz. 2018. *A Tour of Representation Theory.* American Mathematical Society.

[26] I.G. Macdonald. 1998. *Symmetric Functions and Hall Polynomials.* Clarendon Press. https://books.google.de/books?id=srv90XiUbZoC

[27] Stephan Mertens and Cristopher Moore. 2013. The Complexity of the Fermionant and Immanants of Constant Width [Note]. *Theory Comput.* 9 (2013), 273–282. https://doi.org/10.4086/toc.2013.v009a006

[28] T. Mignon and N. Ressayre. 2004. A quadratic bound for the determinant and permanent problem. *International Mathematics Research Notices* 2004, 79 (2004), 4241–4253.

[29] Istvan Miklos and Cordian Riener. 2021. #P-hardness proofs of matrix immanants evaluated on restricted matrices. arXiv:cs.CC/2103.04934

[30] Brendon Rhoades and Mark Skandera. 2006. Kazhdan-Lusztig immanants and products of matrix minors. *Journal of Algebra* 304, 2 (2006), 793–811. https://doi.org/10.1016/j.jalgebra.2005.07.017

[31] B.E. Sagan. 2013. *The Symmetric Group: Representations, Combinatorial Algorithms, and Symmetric Functions.* Springer New York.

[32] I. Schur. 1901. *Über eine Klasse von Matrizen, die sich einer gegebenen Matrix zuordnen lassen.* Ph.D. Dissertation. Friedrich-Wilhelms-Universität zu Berlin.

[33] I. Schur. 1918. Über endliche Gruppen und Hermitesche Formen. *Mathematische Zeitschrift* 1 (1918), 184–207. http://eudml.org/doc/167457

[34] Valery S. Shchesnovich. 2016. The permanent-on-top conjecture is false. *Linear Algebra Appl.* 490 (2016), 196 – 201. https://doi.org/10.1016/j.laa.2015.10.034

[35] Si-Hui Tan, Yvonne Gao, Hubert Guise, and Barry Sanders. 2012. SU(3) Quantum Interferometry with Single-Photon Input Pulses. *Physical Review Letters* 110 (08 2012). https://doi.org/10.1103/PhysRevLett.110.113603

[36] H. N. V. Temperley and Michael E. Fisher. 1961. Dimer problem in statistical mechanics - an exact result. *The Philosophical Magazine: A Journal of Theoretical Experimental and Applied Physics* 6, 68 (1961), 1061–1063. https://doi.org/10.1080/14786436108243366 arXiv:https://doi.org/10.1080/14786436108243366

[37] Leslie G. Valiant. 1979. Completeness Classes in Algebra. In *Proceedings of the 11h Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA*, Michael J. Fischer, Richard A. DeMillo, Nancy A. Lynch, Walter A. Burkhard, and Alfred V. Aho (Eds.). ACM, 249–261. https://doi.org/10.1145/800135.804419

[38] Leslie G. Valiant. 1979. The Complexity of Computing the Permanent. *Theor. Comput. Sci.* 8 (1979), 189–201. https://doi.org/10.1016/0304-3975(79)90044-6