# Counting Matchings with $k$ Unmatched Vertices in Planar Graphs

**Radu Curticapean**[*†]

**Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary**
radu.curticapean@gmail.com

## Abstract

We consider the problem of counting matchings in planar graphs. While *perfect* matchings in planar graphs can be counted by a classical polynomial-time algorithm [26, 33, 27], the problem of counting all matchings (possibly containing unmatched vertices, also known as *defects*) is known to be #P-complete on planar graphs [23].

To interpolate between matchings and perfect matchings, we study the parameterized problem of counting matchings with $k$ unmatched vertices in a planar graph $G$, on input $G$ and $k$. This setting has a natural interpretation in statistical physics, and it is a special case of counting perfect matchings in $k$-apex graphs (graphs that become planar after removing $k$ vertices). Starting from a recent #W[1]-hardness proof for counting perfect matchings on $k$-apex graphs [12], we obtain:

- Counting matchings with $k$ unmatched vertices in planar graphs is #W[1]-hard.
- In contrast, given a plane graph $G$ with $s$ distinguished faces, there is an $\mathcal{O}(2^s \cdot n^3)$ time algorithm for counting those matchings with $k$ unmatched vertices such that all unmatched vertices lie on the distinguished faces. This implies an $f(k,s) \cdot n^{O(1)}$ time algorithm for counting perfect matchings in $k$-apex graphs whose apex neighborhood is covered by $s$ faces.

**1998 ACM Subject Classification** G.2.1 Combinatorics, G.2.2 Graph Theory, F.1.3 Complexity Measures and Classes

**Keywords and phrases** counting complexity, parameterized complexity, matchings, planar graphs

**Digital Object Identifier** 10.4230/LIPIcs.ESA.2016.33

## 1 Introduction

The study of the computational complexity of counting problems was introduced in a seminal paper by Valiant [34] that established the class #P and proved that counting perfect matchings in an unweighted bipartite graph is #P-complete. In a companion paper [35], Valiant proved that counting all (not necessarily perfect) matchings in a graph is #P-complete as well. Even prior to these initial complexity-theoretic results, problems related to matchings and perfect matchings have played an important role in various scientific disciplines.

For instance, the number of perfect matchings in a bipartite graph $G$ arises in enumerative combinatorics and algebraic complexity as the *permanent* of the bi-adjacency matrix associated with $G$ [3, 1]. In statistical physics, counting perfect matchings amounts to evaluating the *partition function* of the *dimer model* [27, 26, 33]: The physical interpretation here is that vertices are discrete points that are occupied by atoms, while edges are interpreted as bonds

[*] Part of this work was carried out while the author was a PhD student at Saarland University in Saarbrücken, Germany, and while he was visiting the Simons Institute for the Theory of Computing in Berkeley, USA. The material also appears in his PhD thesis [10].

[†] The author is supported by the ERC grant PARAMTIGHT, no. 280152.


24th Annual European Symposium on Algorithms (ESA 2016).
Editors: Piotr Sankowski and Christos Zaroliagis; Article No. 33; pp. 33:1–33:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

between the corresponding atoms. The partition function of $G$ is then essentially defined as the number of perfect matchings in $G$, and it encodes thermodynamic properties of the associated system. Likewise, the problem of counting all matchings is known to statistical physicists as the *monomer-dimer model* [23]; in this setting, some points may be unoccupied by atoms. In the intersection of chemistry and computer science, the number of matchings of a graph (representing a molecule) is known as its *Hosoya index* [20].

In view of these applications and the #P-hardness of counting matchings and perfect matchings, several relaxations were considered to cope with these problems. Among these, *approximate* counting and the restriction to *planar* graphs proved most successful. However, once we start incorporating these relaxations, the seemingly very similar problems of counting matchings and counting perfect matchings exhibit stark differences:

- On planar graphs, perfect matchings can be counted in polynomial time by the classical and somewhat marvelous FKT method [27, 26, 33], which reduces this problem to the determinant. The problem of counting all matchings is however #P-complete on planar graphs [23]. In particular, the algebraic machinery in the FKT method breaks down for non-perfect matchings.
- It was shown that the number of matchings in a graph admits a polynomial-time randomized approximation scheme (FPRAS) on general graphs [24]. By a substantial extension of this approach, an FPRAS for counting perfect matchings in bipartite graphs was obtained [25] – but despite great efforts, no FPRAS is known for general graphs.

In the present paper, we focus on the differing behavior of matchings and perfect matchings on planar graphs. To this end, we study the problem #PlanarDefectMatch of counting matchings with $k$ unmatched vertices (which we call $k$-defect matchings) in a planar graph $G$, on input $G$ and $k$. This problem is clearly #P-hard under Turing reductions, as the #P-hard number of matchings in $G$ can be obtained as the sum of numbers of $k$-defect matchings in $G$ for $k = 0, \ldots, |V(G)|$. On the other hand, #PlanarDefectMatch can easily be solved in time $|V(G)|^{\mathcal{O}(k)}$, as we can simply enumerate all $k$-subsets $X \subseteq V(G)$ that represent potential defects, count perfect matchings in the planar graph $G - X$ by the FKT method, and sum up these numbers.

## 1.1 Parameterized counting problems

The fact that #PlanarDefectMatch is #P-hard and polynomial-time solvable for constant $k$ suggests that this problem benefits from the framework of *parameterized counting complexity* [15]. This area is concerned with *parameterized* counting problems, whose instances $x$ come with *parameters* $k$, such as #PlanarDefectMatch or the problem #Clique of counting $k$-cliques in an $n$-vertex graph. Intuitively, the parameterized problem #PlanarDefectMatch considers $k$-defect matchings in planar graphs with $k \ll n$, and the physical interpretation in terms of the monomer-dimer model is that each configuration of the system admits only a small number of "vacant" points that are not occupied by atoms.

Note that both #PlanarDefectMatch and #Clique can be solved in time $n^{\mathcal{O}(k)}$ and are hence in the so-called class XP. One important goal for such problems lies in finding algorithms with running times $f(k) \cdot |x|^{\mathcal{O}(1)}$ for computable functions $f$, which renders the problems *fixed-parameter tractable* (FPT) [15, 16]. If no FPT-algorithms can be found for a given problem, one can try to show its #W[1]-hardness. This essentially boils down to finding a parameterized reduction from #Clique, and it shows that FPT-algorithms for the problem would imply FPT-algorithms for #Clique, which is considered unlikely.

For instance, to prove #W[1]-hardness of #PlanarDefectMatch by reduction from #Clique, we would need to find an algorithm that counts $k$-cliques of an $n$-vertex graph in time

$f(k) \cdot n^{\mathcal{O}(1)}$ with an oracle for #PlanarDefectMatch. Additionally, the algorithm should only invoke the oracle for counting $k'$-defect matchings with $k' \leq g(k)$. Here, both the function $f$ appearing in the running time and the blow-up function $g$ are arbitrary computable functions.

Furthermore, parameterized reductions can also be used to obtain lower bounds under the exponential-time hypothesis #ETH, which postulates that the satisfying assignments to formulas $\varphi$ in 3-CNF cannot be counted in time $2^{o(n)}$ [13, 21, 22]. For instance, it is known that #Clique cannot be solved in time $n^{o(k)}$ unless #ETH fails [5]. If we reduce from #Clique to a target problem by means of a reduction that invokes only blow-up $\mathcal{O}(k)$, then #ETH also rules out $n^{o(k)}$ time algorithms for the target problem [29].

## 1.2 Perfect matchings with planar-like parameters

To put #PlanarDefectMatch into context, let us survey some parameterizations for the problem #PerfMatch of counting perfect matchings and see how these connect to #PlanarDefectMatch.

- The FKT method for planar graphs was extended [18, 30, 12] from planar graphs to graphs of fixed genus $g$, resulting in $\mathcal{O}(4^g \cdot n^3)$ time algorithms for #PerfMatch.
- Polynomial-time algorithms for #PerfMatch were obtained for $K_{3,3}$-free graphs [28, 38] and $K_5$-free graphs [32]. More generally, for every class of graphs excluding a fixed single-crossing minor $H$ (that is, $H$ can be drawn in the plane with at most one crossing), an $f(H) \cdot n^4$ time algorithm is known [7].
- A simple dynamic programming algorithm yields a running time of $3^t \cdot n^{\mathcal{O}(1)}$ for #PerfMatch on graphs of treewidth $t$. By using fast subset convolution [37], the running time can be improved to $2^t \cdot n^{\mathcal{O}(1)}$.

Since all of the tractable classes above exclude fixed minors for fixed parameter values, one is tempted to believe that #PerfMatch could be polynomial-time solvable on *each* class of graphs excluding a fixed minor $H$, and possibly even admit an FPT-algorithm when parameterized by the minimum size of an excluded minor. This last possibility was however ruled out by the following result:[1]

- #PerfMatch is #W[1]-hard on $k$-apex graphs [12]. For $k \in \mathbb{N}$, a graph $G$ is $k$-apex if there is a set $A \subseteq V(G)$ of size $k$ such that $G - A$ is planar. The vertices in $A$ are called *apices*. Since $k$-apex graphs exclude minors on $\mathcal{O}(k)$ vertices, the #W[1]-hardness result for #PerfMatch on $k$-apex graphs implies #W[1]-hardness of #PerfMatch on graphs excluding fixed minors $H$ (when parameterized by the minimum size of such an $H$).

Note that #PerfMatch can be solved in time $n^{\mathcal{O}(k)}$ on $k$-apex graphs by brute-force in a similar way as #PlanarDefectMatch. To cope with the #W[1]-hardness of #PerfMatch in $k$-apex graphs and potentially obtain faster algorithms, we study two special cases:

1. We consider #PlanarDefectMatch, which is indeed a special case, as discussed below.
2. We consider #PerfMatch in $k$-apex graphs whose apices are adjacent with only a bounded number of faces in the underlying planar graph. More in Section 1.4 of the introduction.

## 1.3 From $k$ apices to $k$ defects

To count the $k$-defect matchings in a planar graph $G$, we can equivalently count perfect matchings in the $k$-apex graph $G'$ obtained from $G$ by adding $k$ independent apex vertices adjacent to all vertices of $G$: Every perfect matching of $G'$ then corresponds to a $k$-defect matching of $G$, and likewise, every $k$-defect matching of $G$ corresponds to precisely $k!$ perfect

[1] In fact, recent unpublished work suggests the existence of constant-sized minors $H$ such that #PerfMatch is #P-hard on $H$-minor free graphs.

**Table 1** Counting matchings under different parameterizations and input restrictions

| **counting matchings** | on planar inputs | on general inputs |
|---|---|---|
| with $k$ edges | FPT by [17] | #W[1]-complete by [6, 11] |
| with $k$ defects | #W[1]-hard by Thm. 1 | #P-complete for $k = 0$ by [34] |

matchings of $G'$. This shows that #PlanarDefectMatch reduces to #PerfMatch on $k$-apex graphs, even when the apices in these latter graphs form an independent set and each apex is adjacent with all non-apex vertices. Note that the #W[1]-hardness for the general problem of #PerfMatch on $k$-apex graphs does a priori not carry over to the special case #PlanarDefectMatch, as the edges between apices and the planar graph cannot be assumed to be complete bipartite graphs in the general problem.

Nevertheless, we show in Section 3 that #PlanarDefectMatch is #W[1]-hard. To this end, we reduce from #PerfMatch on $k$-apex graphs by means of a "truncated" polynomial interpolation where we wish to recover only the first $k$ coefficients from a polynomial of degree $n$. The technique is comparable to that used in the first #W[1]-hardness proofs for counting matchings with $k$ edges [2, 6]. Interestingly enough, our reduction maps $k$-apex graphs to instances of counting $k$-defect matchings without incurring any parameter blowup at all. In particular, we obtain the same almost-tight lower bound under #ETH that was known for #PerfMatch on $k$-apex graphs [12].

▶ **Theorem 1.** #PlanarDefectMatch *is* #W[1]*-hard and admits no* $n^{o(k/\log k)}$ *time algorithm unless* #ETH *fails.*

It should be noted that the "primal" problem of counting matchings with $k$ edges is #W[1]-hard on general graphs [6, 11], but becomes FPT on planar graphs [17]. Furthermore, recall that counting matchings with 0 defects (that is, perfect matchings) in general graphs is #P-hard. See also Table 1 for the complexity of counting matchings in various settings.

## 1.4 Few apices that also see few faces

In Section 4, we show that #PerfMatch becomes easier in $k$-apex graphs $G$ when the apex neighborhoods can all be covered by $s$ faces of the underlying planar graph. This setting is motivated by a structural decomposition theorem for graphs $G$ excluding a fixed 1-apex minor $H$: As shown in [14], based on [31], if $G$ excludes a fixed 1-apex minor $H$, then there is a constant $c_H \in \mathbb{N}$ such that $G$ can be obtained by gluing together (in a formalized way) graphs that have genus $\leq c_H$ after removing "vortices" from $\leq c_H$ faces and a set $A$ of $\leq c_H$ apex vertices, whose neighborhood in $G - A$ is however covered by $\leq c_H$ faces. Our setting is a simplification of this general situation as we forbid vortices, gluing, and restrict the genus to 0. We obtain an FPT-algorithm for this restricted case:

▶ **Theorem 2.** *Given as input a graph $G$, a set $A \subseteq V(G)$ of size $k$ and a drawing of $G - A$ in the plane with $s$ distinguished faces $F_1, \ldots, F_s$ such that the neighborhood of $A$ is contained in the union of $F_1, \ldots, F_s$, we can count the perfect matchings of $G$ in time $2^{\mathcal{O}(2^k \cdot \log(k)+s)} \cdot n^4$.*

Note that even with $k = 3$ and $s = 1$, such graphs can have unbounded genus, as witnessed by the graphs $K_{3,n}$ for $n \in \mathbb{N}$: Each graph $K_{3,n}$ is a 3-apex graph whose underlying planar graph (which is an independent set) can be drawn on one single face. However, the genus of $K_{3,n}$ is known to be $\Omega(n)$ [19].

To prove Theorem 2, we first consider a variant of #PlanarDefectMatch where the input graph $G$ is given as a planar drawing with $s$ distinguished faces. The task in this variant is to count $k$-defect matchings such that all defects are contained in the distinguished faces. This problem is FPT, even when $k$ is not part of the parameter.

▶ **Theorem 3.** *Given as input a planar drawing of a graph $G$ with $s$ distinguished faces $F_1, \ldots, F_s$, the following problem can be solved in time $\mathcal{O}(2^s \cdot n^3)$: Count the matchings in $G$ for which every defect is contained in $V(F_1) \cup \ldots \cup V(F_s)$.*

To prove Theorem 3, we implicitly use the technique of combined signatures [12]: Using a linear combination of two planar gadgets from [36], we show that counting the particular matchings needed in Theorem 3 can be reduced to $2^s$ instances of #PerfMatch in planar graphs. We can phrase this result in a self-contained way that does not require the general machinery of combined signatures. It should be noted that the case $s = 1$ was already solved by Valiant [36] and that our proof of Theorem 3 is a rather simple generalization of his construction. In a different context, this idea is also used in [9].

More effort is then required to prove Theorem 2, and we do so by reduction to Theorem 3. To this end, we label each vertex in the planar graph $G - A$ with its neighborhood in the apex set $A$. Each $k$-defect matching in $G - A$ then has a *type*, which is the $k$-element multiset of $A$-neighborhoods of its $k$ defects.[2] We will be able to count $k$-defect matchings $M$ of any specified type among the $(2^k)^k$ possible types, and we observe that the number of extensions from $M$ to a perfect matching in $G$ depends only on its type. This will allow us to recover the number of perfect matchings in $G$.

## 2 Preliminaries

For $n \in \mathbb{N}$, write $[n] = \{1, \ldots, n\}$. Graphs $G$ are undirected and simple. They are unweighted unless specified otherwise. We write $N_G(v)$ for the neighborhood of $v \in V(G)$ in $G$.

### 2.1 Polynomials

We denote the degree of a polynomial $p \in \mathbb{Q}[x]$ by $\deg(p)$. If $\mathbf{x} = (x_1, \ldots, x_t)$ is a list of indeterminates, then we write $\mathbb{N}^{\mathbf{x}}$ for the set of all monomials over $\mathbf{x}$. A *multivariate polynomial* $p \in \mathbb{Q}[\mathbf{x}]$ is a polynomial $p = \sum_{\theta \in \mathbb{N}^{\mathbf{x}}} a(\theta) \cdot \theta$ with $a(\theta) \in \mathbb{Q}$ for all $\theta \in \mathbb{N}^{\mathbf{x}}$, where $a$ has finite support. The polynomial $p$ *contains* a given monomial $\theta \in \mathbb{N}^{\mathbf{x}}$ if $a(\theta) \neq 0$ holds. If $x$ is an indeterminate from $\mathbf{x}$, then we write $\deg_x(p)$ for the *degree of* $x$ in $p$. This is the maximum number $k \in \mathbb{N}$ such that $p$ contains a monomial $\theta$ with factor $x^k$. If $\mathbf{y}$ is a list of indeterminates, then we denote the *total degree of* $\mathbf{y}$ in $p$ as the maximum degree of any monomial $\mathbb{N}^{\mathbf{y}}$ that is contained as a factor of a monomial in $p$.

Furthermore, if $p \in \mathbb{Q}[x, y]$ is a bivariate polynomial and $\xi \in \mathbb{Q}$ is some arbitrary fixed value, we write $p(\cdot, \xi)$ for the result of the substitution $y \leftarrow \xi$ in $p$, and we observe that $p(\cdot, \xi) \in \mathbb{Q}[x]$. Likewise, we write $p(\xi, \cdot)$ for the result of substituting $x \leftarrow \xi$.

### 2.2 (Perfect) matching polynomials

If $G$ is a graph, then a set $M \subseteq E(G)$ of vertex-disjoint edges is called a matching. We write $\mathcal{M}[G]$ for the set of all matchings of $G$. For $M \in \mathcal{M}[G]$, we write $\mathrm{usat}(M)$ for the set of unmatched vertices in $M$. If $|\mathrm{usat}(M)| = k$ for $k \in \mathbb{N}$, we say that $M$ is a $k$-defect matching, and we write $\mathcal{DM}_k[G]$ for the set of $k$-defect matchings of $G$. We also write $\mathcal{PM}[G] = \mathcal{DM}_0[G]$ for the set of perfect matchings of $G$.

[2] This resembles an idea from an algorithm for counting subgraphs of bounded vertex-cover number [11].

If $G$ is an edge-weighted graph with edge-weights $w : E(G) \to \mathbb{Q}$, then we define

$$\#\mathsf{PerfMatch}(G) = \sum_{M \in \mathcal{PM}[G]} \prod_{e \in M} w(e). \tag{1}$$

On planar graphs $G$, we can efficiently compute $\#\mathsf{PerfMatch}(G)$.

▶ **Theorem 4** ([26, 33, 27])**.** *For planar edge-weighted graphs $G$, the value $\#\mathsf{PerfMatch}(G)$ can be computed in time $\mathcal{O}(n^3)$.*

If $G$ is a vertex-weighted graph with vertex-weights $w : V(G) \to \mathbb{Q}$, we define

$$\#\mathsf{MatchSum}(G) = \sum_{M \in \mathcal{M}[G]} \prod_{v \in \mathrm{usat}(M)} w(v). \tag{2}$$

Both $\#\mathsf{PerfMatch}$ and $\#\mathsf{MatchSum}$ are also used in [36]. Note that zero-weights have different semantics in the two expressions: A vertex $v \in V(G)$ with $w(v) = 0$ is required to be matched in all matchings $M \in \mathcal{M}[G]$ that contribute a non-zero term to $\#\mathsf{MatchSum}$. An edge $e \in E(G)$ with $w(e) = 0$ can simply be deleted from $G$ without affecting $\#\mathsf{PerfMatch}(G)$.

Finally, if $X$ is a formal indeterminate, we define the defect-generating matching polynomial of unweighted graphs $G$ as

$$\mu(G) := \sum_{M \in \mathcal{M}[G]} X^{|\mathrm{usat}(M)|} = \sum_{k=0}^{n} \#\mathcal{DM}_k[G] \cdot X^k. \tag{3}$$

Note that $\mu(G) = \#\mathsf{MatchSum}(G')$ when $G'$ is obtained from $G$ by assigning weight $X$ to every vertex of $G$. In this paper, we will be interested in the first $k$ coefficients of $\mu(G)$.

▶ Remark. It is known [4] that for every fixed $\xi \in \mathbb{Q} \setminus \{0\}$, the problem of evaluating $\mu(G; \xi)$ on input $G$ is #P-complete, even on planar bipartite graphs $G$ of maximum degree 3. Note that the evaluation $\mu(G; 0)$ counts the perfect matchings of $G$.

## 2.3 Techniques from parameterized counting

Please consider Section 1.1 for an introduction to parameterized counting complexity, and [15] for a more formal treatment. We write $\leq^T_{fpt}$ for parameterized (Turing) reductions between problems (as introduced in Section 1.1). Furthermore, we write $\leq^{lin}_{fpt}$ for such parameterized reductions that incur only linear parameter blowup, i.e., on instances $x$ with parameter $k$, they only issue queries with parameter $\mathcal{O}(k)$.

Given a universe $\Omega$ and several "bad" subsets of $\Omega$, the inclusion-exclusion principle allows us to count those elements of $\Omega$ that avoid all bad subsets, provided that we know the sizes of intersections of bad subsets.

▶ **Lemma 5.** *Let $\Omega$ be a set and let $A_1, \ldots, A_t \subseteq \Omega$. For $\emptyset \subset S \subseteq [t]$, let $A_S := \bigcap_{i \in S} A_i$ and define $A_\emptyset := \Omega$. Then we have $\left|\Omega \setminus \bigcup_{i \in [t]} A_i\right| = \sum_{S \subseteq [t]} (-1)^{|S|} |A_S|$.*

In applications of Lemma 5, the left-hand side of the equation corresponds to a quantity we wish to determine, while the numbers $|A_S|$ for $S \subseteq [t]$ are computed by oracle calls.

We will also generously use the technique of polynomial interpolation: if a univariate polynomial $p$ has degree $n$ and we can evaluate $p(\xi)$ at $n + 1$ distinct values $\xi$, then we can recover the coefficients of $p$. This can be generalized to multivariate polynomials: If $p$ has $n$ variables, all of maximum degree $d$, and we are given sets $\Xi_1, \ldots, \Xi_n$, all of size $d + 1$, along with evaluations of $p(\xi)$ on all grid points $\xi \in \Xi_1 \times \ldots \times \Xi_n$, then we can determine the coefficients of $p$ in time $\mathcal{O}((d + 1)^{3n})$.

▶ **Lemma 6** ([8]). *Let $p \in \mathbb{Z}[x_1, \ldots, x_n]$ be a multivariate polynomial, and for $i \in [n]$, let the degree of $x_i$ in $p$ be bounded by $d_i \in \mathbb{N}$. Let $\Xi = \Xi_1 \times \ldots \times \Xi_n \subseteq \mathbb{Q}^n$ with $|\Xi_i| = d_i + 1$* for all *$i \in [n]$. Then we can compute the coefficients of $p$ with $\mathcal{O}(|\Xi|^3)$ arithmetic operations when given as input the set $\{(\xi, p(\xi)) \mid \xi \in \Xi\}$.*

# 3 Hardness of #PlanarDefectMatch

We now prove Theorem 1: Given a *planar* graph $G$ and $k \in \mathbb{N}$, it is #W[1]-hard to count the $k$-defect matchings of $G$. This amounts to computing the coefficient of $X^k$ in the matching-defect polynomial $\mu(G)$. We start from the #W[1]-hardness for the following problem #ApexPerfMatch, which follows from Theorem 1.2 and Remark 5.6 in [12]:

▶ **Theorem 7** ([12]). *The following problem #ApexPerfMatch is #W[1]-hard: Compute the value of #PerfMatch$(G)$, when given as input an unweighted graph $G$ and an independent set $A \subseteq V(G)$ of size $k$ such that $G - A$ is planar and each vertex $v \in V(G) \setminus A$ satisfies $|N_G(v) \cap A| \leq 1$. The parameter in this problem is $k$. Furthermore, assuming #ETH, the problem cannot be solved in time $n^{o(k/\log k)}$.*

In the proof of Theorem 1, we introduce an intermediate problem #RestrDefectMatch:

▶ **Problem 8.** The problem #RestrDefectMatch is defined as follows: Given as input a triple $(G, S, k)$ where $G$ is a planar graph, $S \subseteq V(G)$ is a set of vertices, and $k \in \mathbb{N}$ is an integer, count those $k$-defect matchings of $G$ whose defects all avoid $S$, i.e., those $k$-defect matchings $M$ with $S \cap \mathrm{usat}(M) = \emptyset$. The parameter is $k$.

The problem #RestrDefectMatch is equivalent (up to multiplication by a simple factor) to the problem #ApexPerfMatch on graphs $G$ whose apices $A$ are all adjacent to a common subset $S$ of the planar graph $G - A$, and to no other vertices. Our overall reduction then proceeds along the chain

$$\#\mathsf{ApexPerfMatch} \leq^{lin}_{fpt} \#\mathsf{RestrDefectMatch} \leq^{lin}_{fpt} \#\mathsf{PlanarDefectMatch}. \tag{4}$$

## 3.1 From #ApexPerfMatch to #RestrDefectMatch

The first reduction in (4) follows from an application of the inclusion-exclusion principle.

▶ **Lemma 9.** *We have #ApexPerfMatch $\leq^{lin}_{fpt}$ #RestrDefectMatch.*

**Proof of Lemma 9.** We reduce from #ApexPerfMatch and wish to count perfect matchings in an unweighted graph $G$ with apex set $A = \{a_1, \ldots, a_k\}$ and planar base graph $H = G - A$. Note that $A$ is given as part of the input, and it is an independent set. Furthermore, by definition of #ApexPerfMatch, the set $V(H)$ admits a partition into $V_1 \cup \ldots \cup V_k \cup W$ such that all vertices $v \in V_i$ for $i \in [k]$ are adjacent to the apex $a_i$ and to no other apices, while no vertex $v \in W$ is adjacent to any apex. In other words, each vertex $v \in V(H)$ can be colored by its unique adjacent apex, or by a neutral color if $v \in W$.

Recall that $\mathcal{DM}_k[H]$ denotes the set of $k$-defect matchings in $H$. We call a $k$-defect matching $M \in \mathcal{DM}_k[H]$ *colorful* if $|\mathrm{usat}(M) \cap V_i| = 1$ holds for all $i \in [k]$, and we write $\mathcal{C}$ for the set of all such $M$. Note that $\mathrm{usat}(M) \cap W = \emptyset$ for $M \in \mathcal{C}$, since none of its $k$ defects are left over for $W$.

We claim that $\mathcal{PM}[G] \simeq \mathcal{C}$: If $M \in \mathcal{PM}[G]$, then $N = M - A$ satisfies $N \in \mathcal{C}$. Conversely, every $N \in \mathcal{C}$ can be extended to a unique $M \in \mathcal{PM}[G]$ by matching the unique $i$-colored defect to its unique adjacent apex $a_i$.

Given oracle access to #RestrDefectMatch, we can determine $\#\mathcal{C}$ by the inclusion-exclusion principle from Lemma 5: For $i \in [k]$, let $\mathcal{A}_i$ denote the set of those $M \in \mathcal{DM}_k[H]$ whose defects avoid color $i$, i.e., they satisfy $\mathrm{usat}(H, M) \cap V_i = \emptyset$. Then $\mathcal{C} = \mathcal{DM}_k[H] \setminus \bigcup_{i \in [k]} \mathcal{A}_i$.

For $S \subseteq [k]$, write $\mathcal{A}_S = \bigcap_{i \in S} \mathcal{A}_i$ and note that we can compute $\#\mathcal{A}_S$ by an oracle call to #RestrDefectMatch on the instance $(H, \bigcup_{i \in S} V_i, k)$. We can hence compute $\#\mathcal{C} = \#\mathcal{PM}[G]$ via inclusion-exclusion (Lemma 5) and $2^k$ oracle calls to #RestrDefectMatch. ◀

## 3.2 From #RestrDefectMatch to #PlanarDefectMatch

For the second reduction in (4), we wish to solve instances $(G, S, k)$ to #RestrDefectMatch when given only an oracle for counting $k$-defect matchings in planar graphs, *without* the ability of specifying the set $S$. Let $G$, $S$ and $k$ be fixed in the following. Our reduction involves manipulations on polynomials, such as a truncated version of polynomial division:

▶ **Lemma 10.** *Let $X$ be an indeterminate, and let $p, q \in \mathbb{Z}[X]$ be polynomials $p = \sum_{i=0}^{m} b_i X^i$ and $q = \sum_{i=0}^{n} a_i X^i$ with $a_0 \neq 0$. For all $t \in \mathbb{N}$, we can compute $b_0, \dots, b_t$ with $\mathcal{O}(t^2)$ arithmetic operations from $a_0, \dots, a_t$ and the first $t+1$ coefficients of the product $pq$.*

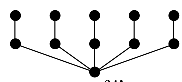
**Proof.** Let $c_0, \dots, c_{n+m}$ enumerate the coefficients of the product $pq$. By elementary algebra, we have $c_i = \sum_{\kappa=0}^{i} a_\kappa b_{i-\kappa}$, which implies the linear system

$$\begin{pmatrix} a_0 & & \\ \vdots & \ddots & \\ a_t & \dots & a_0 \end{pmatrix} \begin{pmatrix} b_0 \\ \vdots \\ b_t \end{pmatrix} = \begin{pmatrix} c_0 \\ \vdots \\ c_t \end{pmatrix}. \tag{5}$$

As this system is triangular with $a_0 \neq 0$ on its main diagonal, it has full rank and can be solved uniquely for $b_0, \dots, b_t$ with $\mathcal{O}(t^2)$ arithmetic operations. ◀

Our proof also relies upon a gadget which will allow to distinguish $S$ from $V(G) \setminus S$.

▶ **Definition 11.** For $\ell \in \mathbb{N}$, an $\ell$-rake $R_\ell$ is a matching $M$ of size $\ell$, together with an additional vertex $w$ adjacent to one vertex of each edge in $M$:




Let $G_{S,\ell}$ be the graph obtained from attaching $R_\ell$ to each $v \in S$. This means adding a local copy of $R_\ell$ to $v$ and identifying the copy of $w$ with $v$. Please note that vertices $v \in V(G) \setminus S$ receive no attachments in $G_{S,\ell}$.

It is obvious that $G_{S,\ell}$ is planar if $G$ is. Recall the defect-generating matching polynomial $\mu$ from (3). We first show that, for fixed $\ell \in \mathbb{N}$, the polynomial $\mu(G_{S,\ell})$ can be written as a weighted sum over matchings $M \in \mathcal{M}[G]$, where each $M$ is weighted by an expression that depends on the number $|\mathrm{usat}(M) \cap S|$. Ultimately, we want to tweak these weights in such a way that only matchings with $|\mathrm{usat}(M) \cap S| = 0$ are counted.

▶ **Lemma 12.** *Define polynomials $r, f_\ell \in \mathbb{Z}[X]$ and $s \in \mathbb{Z}[X, \ell]$ by*

$$r(X) = 1 + X^2, \qquad s(X, \ell) = \ell + 1 + X^2, \qquad f_\ell(X) = (1 + X^2)^{|S|(\ell-1)}.$$

*Then it holds that*

$$\mu(G_{S,\ell}, X) = f_\ell \cdot \sum_{M \in \mathcal{M}[G]} X^{|\mathrm{usat}(M)|} \cdot r^{|S \setminus \mathrm{usat}(M)|} \cdot s^{|S \cap \mathrm{usat}(M)|}. \tag{6}$$

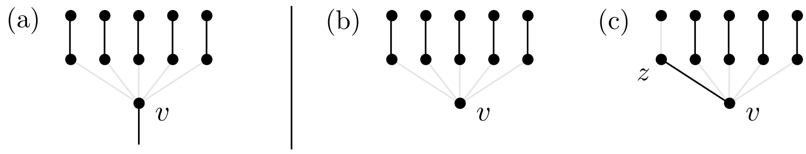



**Figure 1** Possible types of extensions of the rake at $v$. The left case corresponds to $v \notin \mathrm{usat}(M)$, and the two right cases correspond to $v \in \mathrm{usat}(M)$.

**Proof.** Every matching $M \in \mathcal{M}[G]$ induces a certain set $\mathcal{C}_M \subseteq \mathcal{M}[G_{S,\ell}]$ of matchings in $G_{S,\ell}$, where each matching $N \in \mathcal{C}_M$ consists of $M$ together with an extension by rake edges. The family $\{\mathcal{C}_M\}_{M \in \mathcal{M}[G]}$ is easily seen to partition $\mathcal{M}[G_{S,\ell}]$, and we obtain

$$\mu(G_{S,\ell}, X) = \sum_{M \in \mathcal{M}[G]} \underbrace{\sum_{N \in \mathcal{C}_M} X^{|\mathrm{usat}(N)|}}_{=:e(M)}. \tag{7}$$

Every matching $N \in \mathcal{C}_M$ consists of $M$ and rake edges, which are added independently at each vertex $v \in S$. Hence, the expression $e(M)$ in (6) can be computed from the product of the individual extensions at each $v \in S$. To calculate the factor obtained by such an extension, we have to distinguish whether $v$ is unmatched in $M$ or not. The possible extensions at $v$ are also shown in Figure 1.

$v \notin$ **usat**$(M)$ **:** We can extend $M$ at $v$ by any subset of the $\ell$ rake edges not adjacent to $v$, as shown in Figure 1.a. In total, these $2^\ell$ extensions contribute the factor $(1 + X^2)^\ell = (1 + X^2)^{\ell-1} r$.

$v \in$ **usat**$(M)$ **:** We have two choices for extending, shown in the right part of Figure 1: Firstly, we can extend as in the case $v \notin \mathrm{usat}(M)$, and then we obtain the factor $X(1 + X^2)^\ell$. Here, the additional factor $X$ corresponds to the unmatched vertex $v$. This situation is shown in Figure 1.b. Secondly, we can match $v$ to one of its $\ell$ incident rake edges, say to $e = vz$ for a rake vertex $z$, as in Figure 1.c. Then we can choose a matching among the $\ell - 1$ rake edges not incident with $z$. This gives a factor of $\ell X(1 + X^2)^{\ell-1}$. Note that $v$ is matched, but the vertex adjacent to $z$ is not, yielding a factor of $X$.
In total, if $v \in \mathrm{usat}(M)$, we obtain the factor $X(1+X^2)^\ell + \ell X(1+X^2)^{\ell-1} = X(1+X^2)^{\ell-1} s$.

In each matching $N \in \mathcal{C}_M$, every unmatched vertex in $\bar{S} = V(G) \setminus S$ contributes a factor $X$. By multiplying the contributions of all $v \in V(G)$, we have thus shown that

$$\begin{aligned} e(M) &= f_\ell(X) \cdot X^{|\bar{S} \cap \mathrm{usat}(M)|} \cdot r^{|S \setminus \mathrm{usat}(M)|} \cdot (Xs)^{|S \cap \mathrm{usat}(M)|} \\ &= f_\ell(X) \cdot X^{|\mathrm{usat}(M)|} \cdot r^{|S \setminus \mathrm{usat}(M)|} \cdot s^{|S \cap \mathrm{usat}(M)|} \end{aligned}$$

and together with (7), this proves the claim. ◀

Due to the factor $f_\ell$, the expression $\mu(G_{S,\ell})$ is not a polynomial in the indeterminates $X$ and $\ell$. We define a polynomial $p \in \mathbb{Z}[X, \ell]$ by removing this factor.

$$p(X, \ell) := \sum_{M \in \mathcal{M}[G]} X^{|\mathrm{usat}(M)|} \cdot r^{|S \setminus \mathrm{usat}(M)|} \cdot s^{|S \cap \mathrm{usat}(M)|}. \tag{8}$$

Depending upon the concrete application, we will consider $p \in \mathbb{Z}[X, \ell]$ as a polynomial in the indeterminates $\ell$ and $X$, or as a polynomial $p \in (\mathbb{Z}[\ell])[X]$ in the indeterminate $X$ with coefficients from $\mathbb{Z}[\ell]$. In this last case, we write $p = \sum_{i=0}^{n} a_i X^i$ with coefficients $a_i \in \mathbb{Z}[\ell]$ for $i \in \mathbb{N}$ that are in turn polynomials. Then we define

$$[p]_k := \sum_{i=0}^{k} a_i X^i \tag{9}$$

as the restriction of $p$ to its first $k+1$ coefficients. For later use, let us observe the following simple fact about $[p]_k$, considered as a polynomial $[p]_k \in \mathbb{Z}[X, \ell]$.

▶ **Fact 13.** *For $i, j \in \mathbb{N}$, every monomial $\ell^i X^j$ appearing in $[p]_k$ satisfies $i \leq j \leq k$.*

**Proof.** Recall $r$ and $s$ from Lemma 12. The indeterminate $\ell$ appears in $s$ with degree 1, but it does not appear in $r$. In the right-hand side of (8), every term containing a factor $s^t$, for $t \in \mathbb{N}$, also contains the factor $X^t$, because $|S \cap \mathrm{usat}(M)| \leq |\mathrm{usat}(M)|$ trivially holds. Hence, whenever $\ell^i X^j$ is a monomial in $p$, then $i \leq j$. Since the maximum degree of $X$ in $[p]_k$ is $k$ by definition, the claim follows. ◀

In the next lemma, we show that knowing the coefficients of $[p]_k$ allows to solve the instance $(G, S, k)$ to #RestrDefectMatch from the beginning of this subsection. After that, we will show how to compute $[p]_k$ with an oracle for #PlanarDefectMatch.

▶ **Lemma 14.** *Let $\mathcal{N}$ denote the set of (not necessarily $k$-defect) matchings in $G$ with $\mathrm{usat}(M) \cap S = \emptyset$. For all $k \in \mathbb{N}$, we can compute the number of $k$-defect matchings in $\mathcal{N}$ in polynomial time when given the coefficients of $[p]_k$.*

**Proof.** For ease of presentation, assume first we knew *all* coefficients of $p$ rather than only those of $[p]_k$. We will later show how to solve the problem when given only $[p]_k$.

Starting from $p$, we perform the substitution

$$\ell \leftarrow -(1 + X^2) \tag{10}$$

to obtain a new polynomial $q \in \mathbb{Z}[X]$ from $p$. By definition of $s$ (see Lemma 12), we have

$$s(X, -(1 + X^2)) = 0, \tag{11}$$

so every matching $M \notin \mathcal{N}$ has zero weight in $q$. To see this, note that by (8), the weight of each matching $M \in \mathcal{M}[G]$ in $p$ contains a factor $s^{|S \cap \mathrm{usat}(M)|}$. But due to (11), the corresponding term in $q$ is non-zero only if $|S \cap \mathrm{usat}(M)| = 0$. We obtain

$$q = \sum_{M \in \mathcal{N}} X^{|\mathrm{usat}(M)|} \cdot (1 + X^2)^{|S \setminus \mathrm{usat}(M)|}.$$

Since every $M \in \mathcal{N}$ satisfies $|S \setminus \mathrm{usat}(M)| = |S|$, this simplifies to

$$q = (1 + X^2)^{|S|} \cdot \underbrace{\sum_{M \in \mathcal{N}} X^{|\mathrm{usat}(M)|}}_{=:q'} \tag{12}$$

and we can use standard polynomial division by $(1 + X^2)^{|S|}$ to obtain

$$q' = q/(1 + X^2)^{|S|}. \tag{13}$$

By (12), for all $k \in \mathbb{N}$, the coefficient of $X^k$ in $q'$ counts precisely the $k$-defect matchings in $\mathcal{N}$. This finishes the discussion of the idealized setting when all coefficients of $p$ are known.

Recall the three steps involved: The substitution in (10), the polynomial division in (13), and the extraction of the coefficient $X^k$ from $q'$.

The full claim, when only $[p]_k$ rather than $p$ is given, can be shown similarly, but some additional care has to be taken. First, we perform the substitution (10) on $[p]_k$ rather than $p$. This results in a polynomial $b \in \mathbb{Z}[X]$, for which we claim the following:

▶ **Claim 15.** *We have* $[b]_k = [q]_k$.

**Proof.** Let $\Theta_{\leq i}$ for $i \in \mathbb{N}$ denote the set of monomials in $p$ with degree $\leq i$ in $X$. The substitution (10) maps every monomial $\theta$ in the indeterminates $X$ and $\ell$ to some polynomial $g_\theta \in \mathbb{Z}[X]$. Writing $a(\theta) \in \mathbb{Z}$ for the coefficient of $\theta$ in $p$, we obtain $q, b \in \mathbb{Z}[X]$ with

$$q = \sum_{\theta \in \Theta_{\leq n}} a(\theta) \cdot g_\theta, \tag{14}$$

$$b = \sum_{\theta \in \Theta_{\leq k}} a(\theta) \cdot g_\theta. \tag{15}$$

We can conclude that

$$[q]_k \underset{(14)}{=} \left[ \sum_{\theta \in \Theta_{\leq n}} a(\theta) \cdot g_\theta \right]_k = \left[ \sum_{\theta \in \Theta_{\leq k}} a(\theta) \cdot g_\theta \right]_k \underset{(15)}{=} [b]_k\,, \tag{16}$$

where the second identity holds since, whenever $\theta$ has degree $i$ in $X$, for $i \in \mathbb{N}$, then $g_\theta$ contains a factor $X^i$. Hence, for $\theta \in \Theta_{\leq n} \setminus \Theta_{\leq k}$, no terms of the polynomial $g_\theta$ appear in $\left[\sum_{\theta \in \Theta_{\leq n}} a(\theta) \cdot g_\theta\right]_k$. This proves the claim. ◀

Recall the polynomial $q'$ from (13); it remains to apply polynomial division as in (13) to recover $[q']_k$ from $[b]_k$. To this end, we observe that the constant coefficient in $(1+X^2)^{|S|}$ is 1, and that all coefficients of $(1+X^2)^{|S|}$ can be computed by a closed formula. We can thus divide $[b]_k = [q]_k$ by $[(1+X^2)^{|S|}]_k$ via truncated polynomial division (Lemma 10) to obtain $[q']_k$, whose $k$-th coefficient counts the $k$-defect matchings in $\mathcal{N}$, as in the idealized setting discussed before. ◀

Using a combination of truncated polynomial division (Lemma 10) and interpolation, we compute the coefficients of $[p]_k$ with oracle access for #PlanarDefectMatch. This completes the reduction from #RestrDefectMatch to #PlanarDefectMatch.

▶ **Lemma 16.** *We can compute* $[p]_k$ *by a Turing fpt-reduction to* #PlanarDefectMatch *such that all queries have maximum parameter* $k$.

**Proof.** For $\xi$ with $0 \leq \xi \leq k$, let $f_\xi \in \mathbb{Z}[X]$ be the evaluation of the expression $f_\ell$ defined in Lemma 12 at $\ell = \xi$. Define $p_\xi^{(k)} \in \mathbb{Z}[X]$ by

$$p_\xi^{(k)} := [\mu(G_{S,\xi})/f_\xi]_k\,. \tag{17}$$

▶ **Claim 17.** *We have* $p_\xi^{(k)} = [p(\cdot,\xi)]_k = [p]_k(\cdot,\xi)$.

**Proof.** The first identity holds by the definition of $p$ in (8), and by the definition of $p_\xi^{(k)}$. The second identity holds because, for all $t \in \mathbb{N}$, the coefficient of $X^t$ in $p$ is a polynomial in $\ell$ and does not depend on $X$. Hence we may arbitrarily interchange (i) the operation of substituting $\ell$ by expressions not depending on $X$ (and by numbers $\xi \in \mathbb{N}$ in particular), and (ii) the operation of truncating to the first $k$ coefficients. ◀

Recall that $a_t \in \mathbb{Z}[\ell]$ for $t \in \mathbb{N}$ denotes the coefficient of $X^t$ in $p$, which has degree at most $k$ (in the indeterminate $\ell$) by Fact 13. Hence, for fixed $t \in \mathbb{N}$, if we knew the values $a_t(0), \ldots, a_t(k)$, we could recover the coefficients of $a_t \in \mathbb{Z}[\ell]$ via univariate polynomial interpolation. But for $0 \leq \xi, t \leq k$, we can obtain the value $a_t(\xi)$ as the coefficient of $X^t$ in $p_\xi^{(k)}$. This follows from Claim 17. It remains to compute the polynomials $p_0^{(k)}, \ldots, p_k^{(k)}$ with an oracle for #PlanarDefectMatch: First, we observe that the constant coefficient in $f_\xi$ is 1 for all $0 \leq \xi \leq k$, so we can apply the definition of $p_\xi^{(k)}$ from (17) and truncated polynomial division (Lemma 10) to compute $p_\xi^{(k)}$ from $[\mu(G_{S,\xi})]_k$ and $f_\xi$.

It remains only to compute $[\mu(G_{S,\xi})]_k$ and $f_\xi$. Note that the coefficients of $f_\xi$ admit a closed expression by definition, and that $[\mu(G_{S,\xi})]_k$ can be computed by querying the oracle for #PlanarDefectMatch to obtain the number of matchings in $G_{S,\xi}$ with $0, \ldots, k$ defects. ◀

We recapitulate the proof of Theorem 1 in the following.

**Proof of Theorem 1.** By Theorem 7, the problem #ApexPerfMatch is #W[1]-hard, and we have reduced it to #RestrDefectMatch in Lemma 9. By Lemma 16, we can use oracle calls to #PlanarDefectMatch with maximum parameter $k$ to compute the polynomial $[p]_k$, and by Lemma 14, the coefficients of $[p]_k$ allow to recover the solution to #RestrDefectMatch in polynomial time. These two steps establish the second reduction in (4).

Note that both reductions incur only linear blowup on the parameter. Hence, the lower bound of $n^{\Omega(k/\log k)}$ for #ApexPerfMatch under #ETH from Theorem 7 carries over to #PlanarDefectMatch. ◀

## 4 Apices with few adjacent faces

We prove Theorem 2: We present an FPT-algorithm for a restricted version of the problem #PerfMatch on graphs $G$ with an apex set $A$ of size $k$ such that every apex can see only a bounded number of faces. To this end, we first prove a stronger version of Theorem 3 that allows us to compute #MatchSum$(G)$ rather than just count matchings in $G$.

▶ **Theorem 18.** *Assume we are given a drawing of a planar graph $G$ with vertex-weights $w : V(G) \to \mathbb{Q}$ and faces $F_1, \ldots, F_s$ for $s \in \mathbb{N}$ such that all vertices $v \in V(G)$ with $w(v) \neq 0$ satisfy $v \in V(F_1) \cup \ldots \cup V(F_s)$. Then we can compute #MatchSum$(G)$ in time $\mathcal{O}(2^s \cdot n^3)$.*

**Proof.** We first create a partition $B_1, \ldots, B_s$ of $\bigcup_{i\in[s]} V(F_i)$ such that $B_i \subseteq F_i$ for $i \in [s]$ and $B_i \cap B_j = \emptyset$ for $i \neq j$. This can be achieved trivially by assigning each vertex that occurs in several faces $F_i$ to some arbitrarily chosen set $B_i$.

Now we define a type $\theta_M \in \{0,1\}^s$ for each $M \in \mathcal{M}[G]$. For $i \in [s]$, we define

$$\theta_M(i) := \begin{cases} 1 & |\mathrm{usat}(M) \cap B_i| \text{ odd}, \\ 0 & |\mathrm{usat}(M) \cap B_i| \text{ even}. \end{cases}$$

For $\theta \in \{0,1\}^s$, let $\mathcal{M}_\theta[G]$ denote the set of matchings $M \in \mathcal{M}[G]$ with $\theta_M = \theta$, and define

$$S_\theta = \sum_{M \in \mathcal{M}_\theta[G]} \prod_{v \in \mathrm{usat}(M)} w(v).$$

It is clear that #MatchSum$(G) = \sum_{\theta\in\{0,1\}^s} S_\theta$. We show how to compute $S_\theta$ for fixed $\theta$ in time $\mathcal{O}(n^3)$ by reduction to #PerfMatch in planar graphs. For this argument, we momentarily define $\#\mathsf{MatchSum}(G)$ on graphs that have vertex- and edge-weights $w : V(G) \cup E(G) \to \mathbb{Q}$:

$$\#\mathsf{MatchSum}(G) = \sum_{M \in \mathcal{M}[G]} \left( \prod_{v \in \mathrm{usat}(M)} w(v) \right) \left( \prod_{e \in M} w(e) \right).$$

As shown in the proof of Theorem 3.3 in [36], and in Example 15 in [9], for every $t \in \mathbb{N}$, there exist explicit planar graphs $D_t^0$ and $D_t^1$ with $\mathcal{O}(t)$ vertices, which contain special vertices $u_1, \ldots, u_t$ such that all of the following holds:

1. The graphs $D_t^0$ and $D_t^1$ can be drawn in the plane with $u_1, \ldots, u_t$ on their outer faces.
2. Let $H$ be a vertex- and edge-weighted graph with distinct vertices $X = \{v_1, \ldots, v_t\} \subseteq V(H)$ and let $H'$ be obtained from $H$ by placing a disjoint copy of $D_t^0$ into $H$ and connecting $v_i$ to $u_i$ with an edge of weight $w(v_i)$ for all $i \in [t]$. Assign weight 0 to the vertices $v_i$ and to all vertices of $D_t^0$. Then
$$\#\mathsf{MatchSum}(H') = \sum_{\substack{M \in \mathcal{M}[H] \\ |\mathrm{usat}(M) \cap X| \text{ even}}} \left( \prod_{v \in \mathrm{usat}(M)} w(v) \right) \left( \prod_{e \in M} w(e) \right) \tag{18}$$
3. The above statement also applies for $D_t^1$, but the corresponding sum in (18) ranges over those $M \in \mathcal{M}[H]$ where $|\mathrm{usat}(M) \cap X|$ is odd rather than even.

We observe that inserting $D_t^0$ or $D_t^1$ into the face of a planar graph preserves planarity. Hence, we can insert $D_{|B_i|}^{\theta(i)}$ at the vertices $B_i$ along face $F_i$ in $G$, for each $i \in [s]$, and obtain a planar graph $G_\theta$. By construction, we have $\#\mathsf{MatchSum}(G_\theta) = S_\theta$. Furthermore, all vertex-weights in $G_\theta$ are 0 by construction, so we actually have $\#\mathsf{MatchSum}(G_\theta) = \#\mathsf{PerfMatch}(G_\theta)$. Since $G_\theta$ is planar, we can evaluate $\#\mathsf{PerfMatch}(G_\theta)$ in time $\mathcal{O}(n^3)$, thus concluding the proof. ◀

Note that the above theorem allows us to recover the number of $k$-defect matchings in $G$ that have all defects on fixed distinguished faces, for any $k \in \mathbb{N}$: Let $G_X$ be obtained from $G$ by assigning weight $X$ to each vertex. Then $p := \#\mathsf{MatchSum}(G_X)$ is a polynomial of degree at most $n$ and can be interpolated from evaluations $p(0), \ldots p(n)$, but each of these evaluations can be computed in time $\mathcal{O}(2^s \cdot n^3)$ by Theorem 18. As we know, the $k$-th coefficient of $p(X)$ is equal to the number of $k$-defect matchings in $G$.

In the following, we extend this argument by using a variant of multivariate polynomial interpolation (Lemma 6) that applies when we do not require the values of *all* coefficients, but rather only those in a “slice” of total degree $k$, for fixed $k \in \mathbb{N}$. Here, the polynomial $p$ to be interpolated features a distinguished indeterminate $X$, and we wish to extract the coefficient $a_k$ of $X^k$, which is in turn a polynomial. Under certain restrictions, this can be achieved with $f(k) \cdot n$ evaluations, where $n$ denotes the degree of $X$ in $p$.

▶ **Lemma 19.** *Let $p \in \mathbb{Z}[X, \lambda]$ be a multivariate polynomial in the indeterminates $X$ and $\lambda = (\lambda_1, \ldots, \lambda_t)$. Consider $p \in (\mathbb{Z}[\lambda])[X]$ and assume that $p$ has degree $n$ in $X$, and that for all $s \in \mathbb{N}$, the coefficient $a_s \in \mathbb{Z}[\lambda]$ of $X^s$ in $p$ has total degree at most $s$. Let $k \in \mathbb{N}$ be a given parameter, and let $\Xi = \Xi_0 \times \ldots \times \Xi_t \subseteq \mathbb{Q}^{t+1}$ with $|\Xi_0| = n + 1$ and $|\Xi_i| = k + 1$ for all $i > 0$. Then we can compute the coefficients of the polynomial $a_k \in \mathbb{Z}[\lambda]$ with $\mathcal{O}(|\Xi|^3)$ arithmetic operations when given as input the set $\{(\xi, p(\xi)) \mid \xi \in \Xi\}$.*

**Proof.** We consider the grid $\Xi'$ defined by removing the first component from $\Xi$, that is, $\Xi' = \Xi_1 \times \ldots \times \Xi_t$. Observe that $p(\cdot, \xi') \in \mathbb{Z}[X]$ holds for $\xi' \in \Xi'$. Write $\Xi_0 = \{c_0, \ldots, c_n\}$ and note that, for fixed $\xi' \in \Xi'$, our input contains all evaluations

$$p(c_0, \xi'), \ldots, p(c_n, \xi'),$$

so we can use univariate interpolation to determine the coefficient of $X^k$ in $p(\cdot, \xi')$. This coefficient is equal to $a_k(\xi')$ by definition. By performing this process for all $\xi' \in \Xi'$, we can evaluate $a_k(\xi')$ on all $\xi' \in \Xi'$, and hence interpolate the polynomial $a_k \in \mathbb{Z}[\lambda]$ via grid interpolation (Lemma 6). ◀

This brings us closer to the proof of Theorem 2. To proceed, we first consider the case that $A$ is an independent set; the full algorithm is obtained by reduction to this case.

▶ **Lemma 20.** *Let $G$ be an edge-weighted graph, given as input together with an independent set $A \subseteq V(G)$ of size $k$, a planar drawing of $H = G - A$, and faces $F_1, \ldots, F_s$ that contain all neighbors of $A$. Then we can compute $\#\mathsf{PerfMatch}(G)$ in time $k^{\mathcal{O}(2^k)} \cdot 2^{\mathcal{O}(s)} \cdot n^4$.*

▶ Remark. We may assume that every edge $av \in E(G)$ with $a \in A$ and $v \in V(G) \setminus A$ has weight 1: Otherwise, replace $av$ by a path $ar_1r_2v$ with fresh vertices $r_1, r_2$, together with edges $ar_1$ and $r_1r_2$ of unit weight, and an edge $r_2v$ of weight $w(e)$. This clearly preserves the apex number, the value of $\#\mathsf{PerfMatch}$, and ensures that every apex is only incident with unweighted edges.

**Proof.** Recall that $\mathcal{DM}_k[H]$ denotes the set of $k$-defect matchings in $H$. By Remark 4, we can assume that all edges incident with $A$ have unit weight. Let

$$\mathcal{C} = \{M \in \mathcal{DM}_k[H] \mid \mathrm{usat}(M) \subseteq N_G(A)\}.$$

Given any matching $M \in \mathcal{C}$, let $t(M)$ denote its *type*[3], which is defined as the following *multiset* with precisely $k$ elements from $2^A$:

$$t(M) = \{N_G(v) \cap A \mid v \in \mathrm{usat}(M)\}.$$

For the set of all such types, we write $\mathcal{T} = \{t(M) \mid M \in \mathcal{C}\}$ and observe that $|\mathcal{T}| \leq (2^k)^k = 2^{k^2}$. For $t \in \mathcal{T}$, define a graph $S_t$ as follows: Create an independent set $[k]$, corresponding to $A$. Then, for each $N \in t$, create a vertex $v_N$ that is adjacent to all of $N \subseteq [k]$. We note that every *perfect* matching $M \in \mathcal{PM}[G]$ can be decomposed uniquely as $M = B(M) \dot\cup I(M)$ with a $k$-defect matching $B(M) \in \mathcal{C}$ and a perfect matching $I(M) \in \mathcal{PM}[S_{t(B(M))}]$. That is, $B(M) = M - A$ and $I(M) = M[A \cup \mathrm{usat}(B(M))]$. For $t \in \mathcal{T}$, let

$$\begin{aligned} \mathcal{C}_t &= \{M \in \mathcal{C} \mid t(M) = t\}, \\ P_t &:= \sum_{N \in \mathcal{C}_t} \prod_{e \in N} w(e). \end{aligned}$$

It is clear that $\{\mathcal{C}_t\}_{t \in \mathcal{T}}$ partitions $\mathcal{C}$, and this implies

$$\#\mathsf{PerfMatch}(G) = \sum_{t \in \mathcal{T}} P_t \cdot \#\mathsf{PerfMatch}(S_t). \tag{19}$$

To see this, note that each perfect matching of type $t$ can be obtained by extending some matching $M \in \mathcal{C}_t$ (all of which have $k$ defects) by a perfect matching from $\mathrm{usat}(M)$ to $A$, which is precisely a perfect matching of $S_t$. Note that we require here that edges between $\mathrm{usat}(M)$ and $A$ have unit weight, otherwise the graphs $S_t$ would have to be edge-weighted as well and might no longer depend on $t$ only, but would also have to incorporate the edge-weights of $G$.

[3] Please note that these types have no connection to those used in the proof of Theorem 18.

Since $|E(S_t)| \leq k^2$, we can compute $\#\mathsf{PerfMatch}(S_t)$ in time $2^{\mathcal{O}(k^2)}$ by brute force for each $t \in \mathcal{T}$. Hence, we can use (19) to determine $\#\mathsf{PerfMatch}(G)$ in time $|\mathcal{T}| \cdot 2^{\mathcal{O}(k^2)}$ if we know $P_t$ for all $t \in \mathcal{T}$. In the remainder of this proof, we show how to compute $P_t$ by using multivariate polynomial interpolation and the algorithm for $\#\mathsf{MatchSum}$ presented in Theorem 18. To this end, define indeterminates $\lambda = \{\lambda_R \mid R \subseteq A\}$ corresponding to subsets of the apices. Let $X$ denote an additional distinguished indeterminate, and define the following polynomial $p \in \mathbb{Z}[X, \lambda]$. In this definition, we abbreviate $w(M) := \prod_{e \in M} w(e)$.

$$p(X, \lambda) := \sum_{M \in \mathcal{C}} w(M) \cdot X^{|\mathrm{usat}(M)|} \cdot \prod_{v \in \mathrm{usat}(M)} \lambda_{N_G(v) \cap A}. \tag{20}$$

For each type $t \in \mathcal{T}$, say $t = \{N_1, \ldots, N_k\}$, the coefficient of $X^k \cdot \lambda_{N_1} \cdot \ldots \cdot \lambda_{N_k}$ in $p$ is equal to $P_t$. Hence, we can extract $P_t$ for all $t \in \mathcal{T}$ from the coefficients of the monomials in $p$ that have degree exactly $k$ in $X$. Let us denote these monomials by $\mathfrak{N}$, and observe that each monomial $\nu \in \mathfrak{N}$ has total degree $k$ in $\lambda$ by the definition of $p$ in (20).

If we can evaluate $p$ on the elements $(r, \xi)$ from the grid $\Xi = [n+1] \times [k+1]^{2^{|A|}}$, then we can compute the coefficients of all $\nu \in \mathfrak{N}$ in $p$, and thus $P_t$ for all $t \in \mathcal{T}$, by sliced grid interpolation (Lemma 19). Note that $|\Xi| \leq \mathcal{O}(n \cdot k^{2^k})$. We compute these evaluations $p(r, \xi)$ as $p(r, \xi) = \#\mathsf{MatchSum}(H')$, where the vertex-weighted graph $H' = H'(r, \xi)$ is obtained from $H$ via the weight function

$$w(v) := \begin{cases} 0 & \text{if } v \notin N_G(A), \\ r \cdot \xi_{N_G(v) \cap A} & \text{otherwise.} \end{cases}$$

Since all vertices with non-zero weight in $H'$ are contained in the faces $F_1, \ldots, F_s$, we can compute $\#\mathsf{MatchSum}(H')$ in time $\mathcal{O}(2^s \cdot n^3)$ with Theorem 18. We obtain the values $P_t$ for all $t \in \mathcal{T}$, so we obtain $\#\mathsf{PerfMatch}(G)$ via (19) in the required time. ◀

It remains to lift Lemma 20 to the case that $A$ is not an independent set. This follows easily from the fact that, whenever $E(G) = E \dot\cup E'$, then every perfect matching $M \in \mathcal{PM}[G]$ must match every vertex $v \in V(G)$ into exactly one of the sets $E$ or $E'$.

**Proof of Theorem 2.** Let $\mathcal{A} = \mathcal{M}[G[A]]$ denote the set of (not necessarily perfect) matchings of the induced subgraph $G[A]$, and note that $|\mathcal{A}| \leq 2^{k^2}$. For $M \in \mathcal{A}$, let $a_M = \#\mathsf{PerfMatch}(G_M)$, where $G_M$ is defined by keeping from $A$ only $\mathrm{usat}(M)$, and then deleting all edges between the remaining vertices of $A$. We can compute $a_M$ by Lemma 20, since the remaining part of $A$ in $G_M$ is an independent set. It is also easily verified that $\#\mathsf{PerfMatch}(G) = \sum_{M \in \mathcal{A}} a_M \cdot \prod_{e \in M} w(e)$, so we can compute $\#\mathsf{PerfMatch}$ as a linear combination of $2^{k^2}$ values, each of which can be computed by Lemma 20. ◀

**Acknowledgments.** The author wishes to thank Dániel Marx and Holger Dell for pointing out the connection between perfect matchings in $k$-apex graphs and $\#\mathsf{PlanarDefectMatch}$ during the Dagstuhl Seminar 10481 on Computational Counting in 2010. Furthermore, thanks to Mingji Xia for interesting discussions about this topic. In particular, Theorem 3 was found in joint work on combined signatures back in 2013. Thanks also to Markus Bläser for reading earlier drafts of this material as it appeared in my PhD thesis, and thanks to the reviewers of this version for providing helpful comments.

## References


1 Manindra Agrawal. Determinant versus permanent. In *Proceedings of the 25th International Congress of Mathematicians, ICM 2006*, volume 3, pages 985–997, 2006.

2 Markus Bläser and Radu Curticapean. Weighted counting of $k$-matchings is #W[1]-hard. In *IPEC*, pages 171–181, 2012. doi:10.1007/978-3-642-33293-7_17.

3 P. Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory.* Number 7 in Algorithms and Computation in Mathematics. Springer Verlag, 2000. 168 + xii pp.

4 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. A computational proof of complexity of some restricted counting problems. In *TAMC 2009*, pages 138–149, 2009.

5 Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David W. Juedes, Iyad A. Kanj, and Ge Xia. Tight lower bounds for certain parameterized NP-hard problems. *Inf. Comput.*, 201(2):216–231, 2005.

6 Radu Curticapean. Counting matchings of size $k$ is #W[1]-hard. In *ICALP 2013*, pages 352–363, 2013. doi:10.1007/978-3-642-39206-1_30.

7 Radu Curticapean. Counting perfect matchings in graphs that exclude a single-crossing minor. *CoRR*, abs/1406.4056, 2014.

8 Radu Curticapean. Block interpolation: A framework for tight exponential-time counting complexity. In *ICALP 2015*, pages 380–392, 2015.

9 Radu Curticapean. Parity separation: A scientifically proven method for permanent weight loss. *CoRR*, abs/1511.07480, 2015.

10 Radu Curticapean. *The simple, little and slow things count: on parameterized counting complexity.* PhD thesis, Saarland University, 2015.

11 Radu Curticapean and Dániel Marx. Complexity of counting subgraphs: Only the boundedness of the vertex-cover number counts. In *FOCS 2014*, pages 130–139, 2014.

12 Radu Curticapean and Mingji Xia. Parameterizing the permanent: Genus, apices, minors, evaluation mod $2^k$. In *FOCS 2015*, pages 994–1009, 2015.

13 Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlen. Exponential time complexity of the permanent and the Tutte polynomial. *ACM Transactions on Algorithms*, 10(4):21, 2014.

14 Erik D. Demaine, MohammadTaghi Hajiaghayi, and Ken-ichi Kawarabayashi. Approximation algorithms via structural results for apex-minor-free graphs. In *ICALP 2009*, pages 316–327, 2009.

15 Jörg Flum and Martin Grohe. The parameterized complexity of counting problems. *SIAM Journal on Computing*, 33(4):892–922, 2004.

16 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory.* Springer, 2006.

17 Markus Frick. Generalized model-checking over locally tree-decomposable classes. *Theory Comput. Syst.*, 37(1):157–191, 2004.

18 Anna Galluccio and Martin Loebl. On the theory of Pfaffian orientations. I. Perfect matchings and permanents. *Electronic Journal of Combinatorics*, 6, 1998.

19 Frank Harary. *Graph theory.* Addison-Wesley, 1991.

20 Haruo Hosoya. Topological index. A newly proposed quantity characterizing the topological nature of structural isomers of saturated hydrocarbons. *Bulletin of the Chemical Society of Japan*, 44(9):2332–2339, 1971.

21 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

22 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. System Sci.*, 63(4):512–530, 2001.

23 Mark Jerrum. Two-dimensional monomer-dimer systems are computationally intractable. *Journal of Statistical Physics*, 48(1-2):121–134, 1987.

**24** Mark Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM J. Comput.*, 18(6):1149–1178, 1989.

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

**26** Pieter W. Kasteleyn. The statistics of dimers on a lattice: I. The number of dimer arrangements on a quadratic lattice. *Physica*, 27(12):1209–1225, 1961.

**27** Pieter W. Kasteleyn. Graph Theory and Crystal Physics. In *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, 1967.

**28** Charles Little. An extension of Kasteleyn's method of enumerating the 1-factors of planar graphs. In *Combinatorial Mathematics*, LNCS, pages 63–72. Springer, 1974.

**29** Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Lower bounds based on the Exponential Time Hypothesis. *Bulletin of the EATCS*, 84:41–71, 2011.

**30** Tullio Regge and Riccardo Zecchina. Combinatorial and topological approach to the 3d ising model. *Journal of Physics A: Mathematical and General*, 33(4):741, 2000.

**31** Neil Robertson and Paul D. Seymour. Graph minors. XVI. Excluding a non-planar graph. *J. Comb. Theory, Ser. B*, 89(1):43–76, 2003.

**32** Simon Straub, Thomas Thierauf, and Fabian Wagner. Counting the number of perfect matchings in $K_5$-free graphs. In *CCC 2014*, pages 66–77, 2014.

**33** H. N. V. Temperley and Michael E. Fisher. Dimer problem in statistical mechanics – an exact result. *Philosophical Magazine*, 6(68):1478–6435, 1961.

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

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

**36** Leslie G. Valiant. Holographic algorithms. *SIAM J. Comput.*, 37(5):1565–1594, 2008. doi:10.1137/070682575.

**37** Johan M. M. van Rooij, Hans L. Bodlaender, and Peter Rossmanith. Dynamic programming on tree decompositions using generalised fast subset convolution. In *ESA 2009*, pages 566–577, 2009.

**38** Vijay V. Vazirani. NC algorithms for computing the number of perfect matchings in $K_{3,3}$-free graphs and related problems. *Inf. Comput.*, 80(2):152–164, 1989.