Exact and Approximate Pattern Counting in Degenerate Graphs: New Algorithms, Hardness Results, and Complexity Dichotomies

Marco Bressan
Department of Computer Science
University of Milan
Milan, Italy
Email: marco.bressan@unimi.it

Marc Roth
Department of Computer Science & Merton College
University of Oxford
Oxford, United Kingdom
Email: marc.roth@cs.ox.ac.uk

Abstract*—We study the problems of counting the homomorphisms, the copies, and the induced copies of a k-vertex graph H in a d-degenerate n-vertex graph G. By leveraging a new family of graph-minor obstructions called *F-gadgets*, we establish explicit and exhaustive complexity classifications for counting copies and induced copies. For instance, we show that the copies of H in G can be counted in time $f(k,d)\, n^{\max(1,\mathsf{imn}(H))} \log n$, where f is some computable function and $\mathsf{imn}(H)$ is the size of the largest induced matching of H; and that whenever the class of allowed patterns has arbitrarily large induced matchings, no algorithm runs in time $f(k,d)\, n^{o(\mathsf{imn}(H)/\log \mathsf{imn}(H))}$ for any function f, unless the Exponential Time Hypothesis fails. A similar result holds for counting induced copies, with the independence number $\alpha(H)$ in place of $\mathsf{imn}(H)$. These results imply complexity dichotomies, into fixed-parameter tractable versus $\#\mathsf{W}[1]$-hard cases, which parallel the well-known dichotomies when d is not a parameter. Our results also imply the $\#\mathsf{W}[1]$-hardness of counting several patterns, such as k-matchings and k-trees, in d-degenerate graphs. We also give new hardness results and approximation algorithms for generalized pattern counting (i.e., counting patterns with a given property) in degenerate graphs.

***Keywords*-counting problems, degenerate graphs, fine-grained complexity theory, parameterized algorithms**

I. Introduction

We study the following problems. Given a pattern graph H and a host graph G,

- $\#$Sub: compute the number of subgraphs of G isomorphic to H
- $\#$IndSub: compute the number of induced subgraphs of G isomorphic to H
- $\#$Hom: compute the number of homomorphisms from H to G

These three problems arise in a variety of disciplines such as statistical physics [51], [37], database theory [24], [12], [3], constraint satisfaction problems [21], bioinformatics [2], and network analysis [44], [50]. Unfortunately, they are believed to be intractable: loosely speaking, any algorithm for solving them has running time $|G|^{\Theta(k)}$, where $k = |V(H)|$, unless standard conjectures fail. To circumvent this obstacle, it is common to allow the complexity to depend not only on $|G|$ and k, but on other structural parameters as well. For instance, it is well known that $\#$Hom can be solved in time $f(k) \cdot |G|^{\mathrm{tw}(H)}$, for some f, where $k = |V(H)|$ and $\mathrm{tw}(H)$ is the treewidth of H [21]. Since H is typically much smaller than G, a running time of $f(k) \cdot |G|^{O(1)}$ is faster than $|G|^{\Theta(k)}$ even if f grows quickly. Therefore, we can consider $\#$Hom as tractable when restricted to a class C of patterns with bounded treewidth. In this case one says that $\#$Hom(C) is *fixed-parameter tractable*, or FPT for short, where the parameterization is given by k; in other words, it is solvable in time $f(k) \cdot |G|^{O(1)}$. It is also known [21] that, under the Exponential Time Hypothesis (ETH) [35], no algorithm for $\#$Hom(C) runs in time $f(k) \cdot |G|^{o(\mathrm{tw}(H)/\log \mathrm{tw}(H))}$ for any function f, whenever C has unbounded treewidth.[1] Thus, the treewidth of C characterises the fixed-parameter tractability of $\#$Hom(C). Similar characterizations are known for the other two problems as well: Curticapean and Marx [20] have shown that $\#$Sub(C) is FPT if and only if the vertex cover number $\mathsf{vc}(C)$ of C is bounded, while Chen, Thurley and Weyer [15] have shown that $\#$IndSub(C) is FPT if and only if C is finite.

In this work, we study $\#$Sub, $\#$IndSub and $\#$Hom by assuming as additional parameter the *degeneracy* of G, denoted by $d(G)$ or simply d. Informally, G has degeneracy at most d if and only if every subgraph of G has a vertex of degree at most d. We seek to understand when those problems are efficiently solvable given that d is small, or more precisely, when they are solvable in time $f(k,d) \cdot |G|^{O(1)}$ for some function f. More formally, let us denote by $\#\mathrm{Sub}_{\mathrm{D}}$, $\#\mathrm{Hom}_{\mathrm{D}}$, $\#\mathrm{IndSub}_{\mathrm{D}}$ the three problems above, but with the parameterization given by $k + d$. Our goal is to find, for each problem, an

A full version of this paper is available at arxiv.org/abs/2103.05588.

[1] In this paper, when we say that a problem is not FPT, or hard in any other sense, we always assume that one works under standard hardness conjectures (often, ETH or one of its variants). For the sake of readability, from now on we take this assumption for granted, and avoid repeating it.

2575-8454/21/$31.00 ©2021 IEEE
DOI 10.1109/FOCS52979.2021.00036

explicit criterion on the class of allowed patterns C such that the restriction of the problem to C is FPT if and only if C satisfies the criterion. In addition, we aim at establishing fine-grained upper and lower bounds on the complexity of those problems. Degeneracy is a natural notion of sparsity which, unlike treewidth or maximum degree, appears to be essentially constant in many real-world graphs [29], [39]; moreover, the class of graphs with bounded degeneracy is very rich — for instance, it contains every graph family closed under taking minors. For this reason, pattern counting in graphs with small degeneracy has been intensely studied, both in the exact and the approximate variants, with many exciting results [5], [6], [7], [8], [26], [27], [32]. Still, no FPT classification like the one seeked here has been found yet.

The remainder of this extended abstract discusses the related work (Section II), presents our contributions (Section III), discusses our techniques for the upper bounds and the lower bounds (Section IV and Section V), before concluding with some open problems (Section VI).

II. Related Work

For what concerns upper bounds, the seminal work of Chiba and Nishizeki [16] showed that, when H is the complete graph, the copies of H in G can be counted in time $f(k,d) \cdot |V(G)|$. The key idea is that, if G is d-degenerate then we can orient it acyclically so that every node has out-degree at most d. Therefore, in time $O(f(k,d) \cdot |V(G)|)$ we can perform an exhaustive search of depth $k-1$ from each vertex of G. Recently, one of the authors extended this approach to arbitrary patterns via *DAG tree decompositions* [7], [8], which are similar to standard tree decompositions of undirected graphs, but are designed to exploit the degeneracy orientation of G. The resulting running time bounds are in the form $f(k,d) \cdot |V(G)|^{\tau(H)} \cdot \log |V(G)|$, where $\tau(H)$ is called the *dag treewidth* of H (for instance, $\tau(H) = 1$ for the clique). Our upper bounds for exact counting rely on these bounds (for approximate counting, instead, we use novel algorithms). In a line of work on near-linear-time algorithms, Gishboliner et al. [32] and Bera et al. [5], [6] have shown that, if the longest induced cycle of H has size at most 5, then $\#\textsc{Sub}_{\text{D}}$ can be solved in time $f(d,k) \cdot |V(G)| \cdot \log |V(G)|$.

For what concerns hardness results, [7], [8] showed that, unless ETH fails, no algorithm can count the copies of *every* H in time $f(k,d) \cdot |V(G)|^{o(\tau(H)/\log \tau(H))}$, for any f. However, this bound does not apply to *every class* of patterns with unbounded dag-treewidth τ, and so [7], [8] do not establish an FPT classification. Even more recently, [32], [6] showed that, under the Triangle Detection Conjecture,[2] counting the copies of H in G takes time $\Omega(|V(G)|^{1+\gamma})$ for some $\gamma > 0$ whenever H has an induced cycle of length at least 6. Again, this does not establish an FPT classification,[3] and the Triangle Detection Conjecture does not seem sufficiently powerful for that matter. In addition, the techniques used in these works are not immediately extendable to obtain an FPT classification; and although we use a technique of [32] (the extension of complexity monotonicity to degenerate graphs, see below), that technique alone is far from sufficient, and indeed our results are based on substantial novel ingredients.

III. Our Results

We provide several novel results on the fixed-parameter tractability of $\#\textsc{Hom}_{\text{D}}$, $\#\textsc{Sub}_{\text{D}}$, and $\#\textsc{IndSub}_{\text{D}}$. Our main result is a set of upper and lower bounds that establishes a complete and explicit FPT classification for both $\#\textsc{Sub}_{\text{D}}$ and $\#\textsc{IndSub}_{\text{D}}$, therefore giving necessary and sufficient conditions for any class of patterns C to be easily countable (in the FPT sense). In addition, we give hardness results for $\#\textsc{Hom}_{\text{D}}$, as well as for $\#\textsc{IndSub}_{\text{D}}(\Phi)$, the problem of counting the induced copies of all k-vertex patterns that satisfy property Φ. We also give (in)tractability results for the *approximate* versions of these problems, where the goal is to compute a multiplicative ε-approximation of the solution. In particular, we provide algorithms and upper bounds for approximately counting copies and induced copies. Table I and Table II summarize all these results and compares them to the non-degenerate case.

The remainder of this section discusses all the results in detail. Our main claims will be stated both in the language of fine-grained complexity (that is, as fine-grained running time bounds) and in the language of parameterized complexity (that is, as dichotomies between classes of patterns). In what follows, we denote by C a generic computable class of graphs. For every problem, say $\#\textsc{Hom}_{\text{D}}$, we denote by $\#\textsc{Hom}_{\text{D}}(C)$ its restriction to $H \in C$.

A. Counting Subgraphs: The Induced Matching Number

Let $\mathsf{imn}(H)$ be the size of the largest induced matching of H. A class of graphs C has unbounded induced

[2]The conjecture says that, in the word RAM model of $O(\lg n)$ bits, for some $\gamma > 0$ any algorithm deciding whether a graph (V,E) is triangle-free needs $\Omega(|E|^{1+\gamma})$ time in expectation. See [1] for the formal statement.

[3]However, an implicit consequence of [32] is that $\#\textsc{Sub}_{\text{D}}$ is #W[1]-hard when H is a matching, and $\#\textsc{IndSub}_{\text{D}}$ is #W[1]-hard when H is an independent set. Here #W[1] should be considered the parameterized equivalent of #P (see Section 2 in the full version [9] for a formal definition). In fact, we will show that counting k-matchings and counting k-independent sets are the minimal #W[1]-hard cases for $\#\textsc{Sub}_{\text{D}}$ and $\#\textsc{IndSub}_{\text{D}}$ respectively.

Problem (Exact Counting)	**Our results** (parameter: $k+d$)	**Known results** (parameter: k)
$\#\textsc{Sub}$	$f(k,d)\cdot n^{\max(1,\mathsf{imn}(H))}\cdot\log n$ not in $f(k,d)\cdot n^{o(\mathsf{imn}(H)/\log\mathsf{imn}(H))}$ Theorem 1 and Theorem 2	$f(k)\cdot n^{\mathsf{vc}(H)+O(1)}$ not in $f(k)\cdot n^{o(\mathsf{vc}(H)/\log\mathsf{vc}(H))}$ see [20]
$\#\textsc{IndSub}$	$f(k,d)\cdot n^{\alpha(H)}\cdot\log n$ not in $f(k,d)\cdot n^{o(\alpha(H)/\log\alpha(H))}$ Theorem 4 and Theorem 5	$f(k)\cdot n^{k}$ not in $f(k)\cdot n^{o(k/\log k)}$ † see [15]
$\#\textsc{IndSub}(\Phi)$	#W[1]-hard if Φ = connectedness or if Φ is minor-closed, unless trivial or with bounded $\alpha(H)$ Theorem 7 and Theorem 8	see [36], [43], [48] for an overview
$\#\textsc{Hom}$	$f(k,d)\cdot n^{\tau_1(H)}\cdot\log n$ #W[1]-hard if $\mathsf{igm}(H)$ unbounded Theorem 9 and Theorem 10	$f(k)\cdot n^{\mathrm{tw}(H)+1}$ not in $f(k)\cdot n^{o(\mathrm{tw}(H)/\log\mathrm{tw}(H))}$ see [21], [41]

Table I: Our results for the case of *exact* counting. Here imn, vc, α, τ_1, tw, igm denote respectively: induced matching number, vertex-cover number, independence number, dag treewidth, treewidth, and size of the largest induced grid minor. All fine-grained lower bounds assume the ETH. All the hardness results hold for any computable class C of patterns whose relevant parameter (e.g., imn) is unbounded. The fine-grained lower bound for counting homomorphisms was shown in [41] and transfers to subgraphs and induced subgraphs via complexity monotonicity [19].

† For dense classes of graphs C, the conditional lower bound is known to be tight (cf. [13], [14], [47]).

matching number if for every $c \in \mathbb{N}$, there exists a graph $H \in C$ with $\mathsf{imn}(H) > c$. We prove:

Theorem 1. *There exists a computable function f such that $\#\textsc{Sub}$ can be solved in time*[4]

$$f(|H|, d(G)) \cdot |V(G)|^{\max(1,\mathsf{imn}(H))} \cdot \log |V(G)| .$$

Moreover, for any computable class of graphs C of unbounded induced matching number, there is no function f such that $\#\textsc{Sub}(C)$ can be solved in time

$$f(|H|, d(G)) \cdot |V(G)|^{o(\mathsf{imn}(H)/\log\mathsf{imn}(H))} ,$$

unless ETH fails.

In the language of parameterized complexity, Theorem 1 yields:

Theorem 2. *Let C be a computable class of graphs. If C has bounded induced matching number, then $\#\textsc{Sub}_{\mathrm{D}}(C)$ is* FPT. *Otherwise, $\#\textsc{Sub}_{\mathrm{D}}(C)$ is #W[1]-hard.*

Our upper bounds subsume several known algorithms for subgraph counting in degenerate graphs. For instance, we subsume the clique-counting algorithm of Chiba and Nishizeki [16] or the (k,k)-biclique counting algorithm of Eppstein [28] — indeed, both cliques and bicliques have induced matching number 1. Our negative results subsume #W[1]-hardness of counting k-matchings (implicit in [32]) and k-cycles (see [33, Footnote 4]), and imply novel conditional lower bounds for a variety of patterns such as paths, triangle-packings, and trees:

Corollary 3. *The following problems are #W[1]-hard when parameterized by k and $d(G)$ and cannot be solved in time $f(k, d(G)) \cdot |G|^{o(k/\log k)}$ for any function f, unless ETH fails:*

1) *compute the number of k-cycles in G.*
2) *compute the number of k-paths in G.*
3) *compute the number of k-matchings in G.*
4) *compute the number of k-triangle-packings (i.e., disjoint union of k triangles) in G.*
5) *compute the number of k-trees (i.e., trees with k vertices) in G.*[5]

Before, #W[1]-hardness and fine-grained lower bounds for these patterns were known only for non-degenerate host graphs [31], [17], [19].

[4]The factor $\log|V(G)|$ can be reduced to $O(1)$ if we accept a bound in expectation, by replacing the dictionary with deterministic logarithmic access time of [8] with a dictionary with expected access time $O(1)$.

[5]While the case of counting k-trees does not follow directly from Theorems 1 and 2, we will see that it follows from their proofs.

Problem (Approx. Counting)	**Our results** (parameter: $k+d$)	**Known results** (parameter: k)
#SUB	$f(k,d)\cdot(1/\varepsilon)^2\cdot n^{\tau_1(H)+o(1)}$ Theorem 12	$k^{O(k)}\cdot(1/\varepsilon)^{O(1)}\cdot n^{\mathsf{tw}(H)+O(1)}$ see [4]
#INDSUB	$(kd)^{O(k)}\cdot(1/\varepsilon)^2\cdot n^{\mathsf{imn}(H)+1+o(1)}$ Theorem 13	W[1]-hard for any infinite class of patterns‡ see [15]
#INDSUB(Φ)	FPT if Φ is minor-closed Theorem 14	see [43] for an overview

Table II: Our results for the case of *approximate* counting. The symbols tw, τ_1, and imn are as in Table I. Note that #HOM is left out since we cannot prove bounds better than those for exact counting; in fact, even in the non-degenerate case it is not known whether approximate counting is easier than exact counting, see [11].

‡ The problem of detecting an induced copy of a pattern graph H in G is W[1]-hard (when parameterized by $k = |V(H)|$), whenever the class of allowed pattern graphs is infinite [15]. Since an ε-approximating reveals whether the solution is zero, the computation of the latter is W[1]-hard under randomised reductions.

B. Counting Induced Subgraphs: The Independence Number

Let $\alpha(H)$ be the size of the largest independent set of H. A class of graphs C has unbounded independence number if for every $c \in \mathbb{N}$, there exists a graph $H \in C$ with $\alpha(H) > c$. We prove:

Theorem 4. *There exists a computable function f such that #INDSUB can be solved in time*[6]

$$f(|H|, d(G)) \cdot |V(G)|^{\alpha(H)} \cdot \log |V(G)| .$$

Moreover, for any computable class of graphs C of unbounded independence number, there is no function f such that #INDSUB(C) can be solved in time

$$f(|H|, d(G)) \cdot |V(G)|^{o(\alpha(H)/\log \alpha(H))} ,$$

unless ETH fails.

In the language of parameterized complexity, Theorem 4 yields:

Theorem 5. *Let C be a computable class of graphs. If C has bounded independence number, then #INDSUB$_{\mathrm{D}}(C)$ is* FPT. *Otherwise, #INDSUB$_{\mathrm{D}}(C)$ is #W[1]-hard.*

As a consequence of our results, we obtain novel conditional lower bounds for several patterns:

Corollary 6. *The following problems are #W[1]-hard when parameterized by k and $d(G)$ and cannot be solved in time $f(k, d(G)) \cdot |G|^{o(k/\log k)}$ for any function f, unless ETH fails.*

1) *compute the number of induced copies of the (k,k)-biclique in G.*
2) *compute the number of k-independent sets in G.*
3) *compute the number of induced k-cycles in G.*
4) *compute the number of induced k-paths in G.*
5) *compute the number of induced k-matchings in G.*

Remark. Between our upper and lower bounds, there is a gap in the exponent due to the factors $(\log \mathsf{imn}(H))^{-1}$, $(\log \alpha(H))^{-1}$, and $(\log k)^{-1}$. This is not an artifact of our analysis, but arises from the well-known "can you beat treewidth" open problem, which asks whether the standard treewidth-based dynamic programming algorithms for finding and counting homomorphisms are optimal (Conjecture 1.3 in [42]). If they are, that is, if "you cannot beat treewidth", as recent results seem to suggest [38], [10], then all those factors can be dropped and the exponents of our upper bounds are asymptotically tight.

C. Generalised Induced Subgraph Counting

Following Jerrum and Meeks [36], we study the following problem. Let Φ be a fixed graph property. The problem #INDSUB(Φ)[7] expects as input a graph G and a positive integer k, and the goal is to compute the number of induced subgraphs of size k in G that satisfy Φ. The parameterization is given by k. Initially, Jerrum and Meeks established #INDSUB(Φ) to be (#W[1]-)hard if Φ is the property of being connected. Today, it is

[6] Again, the $\log |V(G)|$ factor can be removed if we want only an *expected* running time bound.

[7] "p-#UNLABELLEDINDUCEDSUBGRAPHSWITHPROPERTY(Φ)" in [36].

conjectured that $\#\textsc{IndSub}(\Phi)$ is intractable whenever Φ is non-trivial, and recent results go in this direction [48].

In principle, one can hope that $\#\textsc{IndSub}(\Phi)$ becomes easier when d is small. However we prove that, even in this case, the problem remains mostly intractable. As usual, let $\#\textsc{IndSub}_{\text{D}}(\Phi)$ be the version of $\#\textsc{IndSub}(\Phi)$ with $d(G)$ as additional parameter. As a first result, we prove:

Theorem 7. *Let Φ be the property of being connected. Then $\#\textsc{IndSub}_{\text{D}}(\Phi)$ is $\#\mathsf{W}[1]$-hard.*

This is only a special case, and understanding the complexity of $\#\textsc{IndSub}_{\text{D}}(\Phi)$ for any Φ remains elusive, even for the not necessarily degenerate case. However, we provide a complete picture in case of *minor-closed properties*. This is a well-studied class that includes, for example, planarity and acyclicity.

Theorem 8. *Let Φ be a minor-closed property. If Φ is trivial (i.e., constant) or of bounded independence number, then $\#\textsc{IndSub}_{\text{D}}(\Phi)$ is FPT. Otherwise $\#\textsc{IndSub}_{\text{D}}(\Phi)$ is $\#\mathsf{W}[1]$-hard.*

D. Counting Homomorphisms: The Source of Hardness

Recall that $\#\textsc{Hom}_{\text{D}}$ is the problem of counting homomorphisms, parameterized by $k + d(G)$. A central part of our work consists in novel hardness results for $\#\textsc{Hom}_{\text{D}}$. This is indeed the starting point for all the hardness results for $\#\textsc{Sub}_{\text{D}}$ and $\#\textsc{IndSub}_{\text{D}}$ that we described above. From a technical standpoint, our main finding is that induced grid minors[8] are obstructions:

Theorem 9. *Let C be a computable class of graphs. If C has induced grid minors of unbounded size, then $\#\textsc{Hom}_{\text{D}}(C)$ is $\#\mathsf{W}[1]$-hard.*

Notice the parallel with $\#\textsc{Hom}(C)$, which is $\#\mathsf{W}[1]$-hard if C has grid minors of unbounded size, *where the minors are not necessarily induced* (this follows by the results of [21] and by the Excluded Grid Theorem [45]). However, while $\#\textsc{Hom}(C)$ is known to be hard *if and only if* C has grid minors of unbounded size, for $\#\textsc{Hom}_{\text{D}}(C)$ we give only the "if" direction. Hence, we do not know if $\#\textsc{Hom}_{\text{D}}(C)$ is FPT when C has bounded induced grid minors. We leave closing this gap as an open problem, see Section VI.

Even though we do not prove that $\#\textsc{Hom}_{\text{D}}$ is FPT when C has bounded induced grid minors, we give some positive results for $\#\textsc{Hom}_{\text{D}}$. In particular, we prove upper bounds based on the cliquewidth of the *skeleton graph* of the acyclic orientations of H. Given an acyclic orientation $H_{\rightarrow}$ of H, its skeleton graph $\Lambda(H_{\rightarrow})$ is the bipartite directed graph having on the left side the *sources* of $H_{\rightarrow}$ (vertices with no incoming arcs), and on the right side the *joints* of $H_{\rightarrow}$ (vertices reachable from two or more sources), and where the arc (s, v) exists if and only if v is reachable from s.[9] The skeleton clique-width $\mathsf{skel\text{-}cw}(H)$ of H is the maximum clique-width of $\Lambda(H_{\rightarrow})$ over all acyclic orientations of H. Our result is:

Theorem 10. *There exists a computable function f such that $\#\textsc{Hom}$ can be solved in time*

$$f(|H|, d(G)) \cdot |V(G)|^{\mathsf{skel\text{-}cw}(H)} \cdot \log |V(G)| .$$

E. A Remarkable Parallel

The hardness results described above reveal a remarkable parallel between pattern counting in general graphs and pattern counting in bounded-degeneracy graphs: in both cases, a class of patterns C is hard to count if C is not free from some specific minor, with the only difference that, in the degenerate case, the minor must be induced. This minor-free condition actually gives a dichotomy into $\#\mathsf{W}[1]$-hard and FPT cases, with the possible exception of $\#\textsc{Hom}_{\text{D}}$, for which we know only the hardness direction.

Lemma 11. *Let $P \in \{\#\textsc{Hom}, \#\textsc{Sub}, \#\textsc{IndSub}\}$, and let P_{D} be the version of P parameterized by $k + d(G)$. Then $P(C)$ and $P_{\text{D}}(C)$ are $\#\mathsf{W}[1]$-hard if C has, respectively, minors and induced minors of unbounded size, where the minors are:*

- *for $P = \#\textsc{Hom}$, grids*
- *for $P = \#\textsc{Sub}$, matchings*
- *for $P = \#\textsc{IndSub}$, independent sets*

With the possible exception of $\#\textsc{Hom}_{\text{D}}$, the converse holds too, with the problem becoming FPT.

This parallel suggests, of course, that induced grid minors are the right obstructions to $\#\textsc{Hom}_{\text{D}}$. This is the main problem that our work leaves open, see Section VI.

F. Approximate Counting

Since for exact pattern counting our results are mostly negative, we also consider *approximate* pattern counting. In particular, we study fixed-parameter tractable randomised approximation schemes (FPTRASes, [4]), which are the FPT equivalent of fully polynomial-time randomised approximation schemes. Let P be a counting problem with parameterization κ. For any instance I of P, let $P(I)$ be the corresponding solution. An ε-*approximation* of $P(I)$ is any rational number $\hat{c}$ such that $(1-\varepsilon) \cdot P(I) \leq \hat{c} \leq (1+\varepsilon) \cdot P(I)$. An FPTRAS

[8]A graph F is an induced minor of a graph H if F can be obtained from H by deleting vertices and contracting edges. In contrast to not necessarily induced minors, edge-deletions are not allowed.

[9]The skeleton graph was introduced in [7] to prove that the dag-treewidth of any pattern H is at most $\lfloor \frac{k}{4} \rfloor + 2$, and captures the structure of $H_{\rightarrow}$ that is relevant for the DAG tree decomposition.

for P and κ is a (randomised) algorithm $\mathbb{A}$ that, on input I and ε, returns an ε-approximation of $P(I)$ with probability at least $2/3$, and has running time bounded by $f(\kappa(I)) \cdot \mathrm{poly}(|I|, \varepsilon^{-1})$. The error probability $2/3$ can be improved by standard probability amplification to any $\delta > 0$, at the cost of an additional factor of $O(\log(1/\delta))$ in the running time.

We begin with the approximate counting of copies. In the general case, Arvind and Raman [4] show that $\#\textsc{Sub}(C)$ admits an FPTRAS if C has bounded treewidth. Here we show that, in the bounded-degeneracy case, the treewidth can be replaced by the dag-treewidth:

Theorem 12. *There exists a computable function f such that $\#\textsc{Sub}$ can be ε-approximated with probability $2/3$ in time $\varepsilon^{-2} \cdot f(k,d) \cdot n^{\tau_1(H)+o(1)}$. Hence, $\#\textsc{Sub}_{\mathrm{D}}(C)$ has an FPTRAS whenever $\tau_1(C)$ is bounded.*

Combined with our hardness results, this implies that there are classes of patterns for which, in degenerate graphs, exact counting is hard but approximate counting is easy. For example, let C be the class of all k-wreath graphs.[10] One can see that C has unbounded induced matching number, and therefore by Theorem 2 $\#\textsc{Sub}_{\mathrm{D}}(C)$ is $\#\mathsf{W}[1]$-hard; but its dag treewidth turns out to be at most 2, hence by Theorem 12 $\#\textsc{Sub}_{\mathrm{D}}(C)$ has an FPTRAS. Moreover, C has unbounded treewidth, and thus Theorem 12 is not subsumed by [4]. Further classes with bounded dag treewidth but unbounded induced matching number and unbounded treewidth include, for example, exploded paths and unions of cliques.

Next, we consider approximate counting of induced copies. In the general case, efficient approximation algorithms are unlikely to exist, as even *detecting* an induced copy is $\mathsf{W}[1]$-hard [15]. In sharp contrast we show that, in the degenerate case, FPTRASes exist whenever the patterns have bounded induced matching number:

Theorem 13. *The problem $\#\textsc{IndSub}$ can be ε-approximated with probability $2/3$ in time*

$$O(\varepsilon^{-2} k^{O(k)} d^{k+1} \cdot |V(G)|^{\mathsf{imn}(H)+1+o(1)}).$$

Hence, $\#\textsc{IndSub}_{\mathrm{D}}(C)$ has an FPTRAS whenever $\mathsf{imn}(C)$ is bounded.

Also in this case, there are classes of patterns for which exact counting is hard but approximate counting is tractable. For instance, consider the class C of all bicliques, for which $\alpha(C)$ is unbounded but $\mathsf{imn}(C) = 1$: exact counting is $\#\mathsf{W}[1]$-hard by Theorem 5, yet by Theorem 13 we have an FPTRAS with running time proportional to $n^{2+o(1)}$.

Finally, we consider the generalised subgraph counting problem, $\#\textsc{IndSub}_{\mathrm{D}}(\Phi)$, see Section III-C. By Theorem 8, $\#\textsc{IndSub}_{\mathrm{D}}(\Phi)$ is $\#\mathsf{W}[1]$-hard for any non-trivial minor-closed property of unbounded independence number. Here we complement that result by showing that, for minor-closed properties, the approximate version is easy:

Theorem 14. *Let Φ be a minor-closed graph property. Then $\#\textsc{IndSub}_{\mathrm{D}}(\Phi)$ has an FPTRAS.*

We conclude with a note about hardness of approximate counting in bounded-degeneracy graphs. In the full version [9], we show hardness for some classes of patterns (for example, the subdivisions of a clique). Unfortunately, those hardness results do not match the upper bounds above, so we do not show a dichotomy like we did for exact counting. However we point out that, for approximate counting problems, complexity dichotomies are rare, and sometimes unlikely to exist [25]. In fact, even the case of arbitrary host graphs, which has been studied for a long time, is not resolved yet: While the algorithm of Arvind and Raman [4] establishes tractability for H of bounded treewidth, it is open whether the remaining instances yield hardness. On the positive side, we provide a possible starting point for future research in Section VI.

IV. Overview of our Techniques: Upper Bounds

For exact counting, we use the dynamic programming algorithm of [7], [8] based on *dag tree decompositions*. The algorithm has running time in the form $f(d,k) \cdot n^{\tau_i(H)} \log n$, where $\tau_1(H) \le \tau_2(H) \le \tau_3(H)$ are three variants of the *dag treewidth* of H; τ_1 is for counting homomorphisms, τ_2 is for counting copies, and τ_3 is for counting induced copies. Our contribution here is to prove upper bounds on τ_1, τ_2, τ_3 as a function of imn and α. We also bound τ_1, τ_2, τ_3 as a function of more complex parameters, see the full version [9]. The techniques here are mostly constructive: we show how to build a dag tree decomposition of H starting from (say) the tree decomposition of certain induced minors of H, see below.

For approximate counting, we make use of a recent result [22] that reduces parameterized approximate counting problems to their *colourful* decision version. In the colourful decision version, G is given with a (not necessarily proper) k-colouring of its vertices, and we are asked whether it contains or not a colourful copy of H (one that spans all k colours). The result of [22] says that, if one can solve this problem in time $T(G,k,d)$, then in time $\varepsilon^{-2} k^{2k} n^{o(1)} T(G,k,d)$ one can

[10] For $k \ge 3$, the graph W_k has vertices $V_0 \,\dot\cup \ldots \dot\cup\, V_{k-1}$ and, for all $i \in \{0,\ldots,k-1\}$ it contains all edges between V_i and $V_{i+1 (\mathsf{mod}\ k)}$.

probabilistically count the (uncoloured) copies of H in G within a multiplicative error of ε. This holds for both subgraphs and induced subgraphs. Hence, our contribution here is to give FPT algorithms for the colourful decision version of our counting problems. For example, to decide if G contains a colourful independent set, we proceed as follows: If every colour class V_i of G is large, say $|V_i| > kd$, then G necessarily contains a colourful k-independent set (this can be seen by sorting G in degeneracy ordering and greedily removing the first vertex of each colour). Otherwise, if there are $\ell \geq 1$ colour classes of size $\leq kd$, then we can explicitly enumerate their colourful ℓ-independent sets in time $O((kd)^\ell)$ and recurse on the remaining classes. Our algorithms use this routine and other FPT-style arguments.

V. Overview of our Techniques: Lower Bounds

Most of our work is devoted to the lower bounds. We prove those bounds by (i) developing hardness results for $\#\textsc{Hom}_\text{D}$, and (ii) lifting them to $\#\textsc{Sub}_\text{D}$ and $\#\textsc{IndSub}_\text{D}$. These two steps are carried out with different techniques:

A novel obstruction: F-gadgets. In the first step we introduce F-gadgets, a novel kind of obstructions designed to capture the hardness of counting homomorphisms in bounded-degeneracy graphs. Informally, a graph H has an F-gadget if H contains an induced subgraph that can be obtained from F by substituting each vertex v by a connected component S_v and each edge e by a path P_e of length at least 2 (see Figure 1 for an example, and the full version [9] for a formalization). Therefore, F-gadgets can be seen as induced minors under constraints, and indeed, if F is an F-gadget of H, then F is an induced minor of H, but the converse is not true (for instance, the complete graph is an induced minor of itself, but not an F-gadget of itself).

Exploiting F-gadgets, we prove the following result: if a computable class of graphs C has F-gadgets of unbounded treewidth, then $\#\textsc{Hom}_\text{D}(C)$ is $\#\mathsf{W}[1]$-hard. To this end, we show that the problem of counting the homomorphisms from a graph F to an arbitrary graph G can be FPT-reduced to the problem of counting the homomorphisms from a graph H to a $|H|$-degenerate graph G', where H has an F-gadget and $|G'| = \text{poly}(|G|)$. More precisely, let $\mathcal{F}(C)$ be the class of all graphs F such that some $H \in C$ has an F-gadget. We show the following paramerized Turing reduction:

$$\#\textsc{Hom}(\mathcal{F}(C)) \leq_\text{T}^\text{fpt} \#\textsc{Hom}_\text{D}(C) \quad (1)$$

Since $\#\textsc{Hom}(\mathcal{F}(C))$ is $\#\mathsf{W}[1]$-hard whenever $\mathcal{F}(C)$ has unbounded treewidth [21], it follows that $\#\textsc{Hom}_\text{D}(C)$ is $\#\mathsf{W}[1]$-hard whenever C has F-gadgets of unbounded treewidth, as we claimed. For technical reasons, we perform the reduction using the *colour-prescribed* variants of the problems, where G is "coloured" with the vertices of the pattern[11], and we must count only the homomorphisms that respect that colouring. The link with the original (i.e., uncoloured) problems are given by standard results, see [21], [34], [23] as well as [18], [46].

Although F-gagdets are the technical machinery used in the proof, our hardness result can be restated in more natural terms using *induced grid minors*, see Theorem 9. To this end, we establish the following connection:

Lemma 15. *A class of graphs C has F-gadgets of unbounded treewidth if and only if it has unbounded induced grid minors.*

Therefore, our hardness result can be rephrased as follows: $\#\textsc{Hom}_\text{D}(C)$ is $\#\mathsf{W}[1]$-hard for any computable class of graphs C having unbounded induced grid minors.

Complexity monotonicity and expanders. In the second step, we use *complexity monotonicity*, a principle discovered independently by Curticapean, Dell and Marx [19] and by Chen and Mengel [12]. It is well-known (cf. [40, 5.2.3]) that subgraph counts can be written as finite linear combinations of homomorphism counts, that is:

$$|\mathsf{Sub}(H \to G)| = \sum_{H'} a_H(H') \cdot |\mathsf{Hom}(H' \to G)|\,, \quad (2)$$

where H' ranges over all possible graphs, but a_H is non-zero only for a finite number of them.[12] Therefore, computing $|\mathsf{Sub}(H \to G)|$ is *at most as hard* as computing the terms $|\mathsf{Hom}(H' \to G)|$. Now, complexity monotonicity says that computing $|\mathsf{Sub}(H \to G)|$ is *precisely as hard as* computing the single hardest term $|\mathsf{Hom}(H' \to G)|$ for which $a_H(H') \neq 0$. This allows one to prove hardness results for $\#\textsc{Sub}$ starting from hardness results for $\#\textsc{Hom}$. This principle was originally shown for general graphs G, but Gishboliner et al. [32] showed that it holds when G is degenerate as well, providing a way to link the hardness of $\#\textsc{Sub}_\text{D}$ to that of $\#\textsc{Hom}_\text{D}$. The same principle also holds for computing *induced* subgraph counts.

To exploit this principle, we show the following result. For any graph H, there exists some H' such that $a_H(H') \neq 0$ and that H' has an F-gadget that is a regular expander of treewidth $\Omega(\mathsf{imn}(H))$. Since $\#\textsc{Hom}_\text{D}(C)$ is hard when $\text{tw}(\mathcal{F}(C))$ is unbounded, by complexity monotonicity we infer that $\#\textsc{Sub}_\text{D}(C)$ is

[11] More precisely, the colouring is a homomorphism from G to the pattern, see Section 2 in the full version [9].

[12] Note that a_H is a function only depending on H (and not on G). In particular, for every fixed H and H', $a_H(H')$ is a constant.

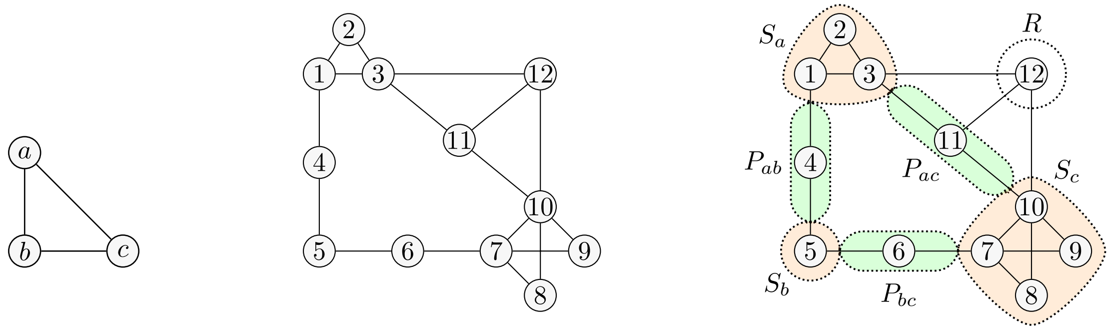

Figure 1: Left to right: a graph F, a graph H, and an F-gadget of H.

hard when $\mathsf{imn}(C)$ is unbounded. Besides mere hardness, we also make sure to preserve the parameters along the reductions, which yields us the fine-grained lower bounds of Theorem 1. The argument for $\#\textsc{IndSub}_{\mathrm{D}}(C)$ is the same, but in this case the F-gadget turns out to be a regular expander of treewidth $\Omega(\alpha(H))$, which yields the fine-grained lower bounds of Theorem 4.

We use the same strategy for $\#\textsc{IndSub}_{\mathrm{D}}(\Phi)$. However, this requires significantly more work: given only Φ, or some property enjoyed by Φ (for instance, given only that Φ is minor-closed), it is much harder to determine which coefficients vanish in the linear combination of a specific pattern H that satisfies Φ. To do this, we rely on the "algebraic approach to hardness" [23]. Let $|\mathsf{IndSub}(\Phi, k \to \star)|$ be the function that maps G to the number of k-vertex induced subgraphs of G that satisfy Φ. First, we can again write $|\mathsf{IndSub}(\Phi, k \to \star)|$ as a linear combination of homomorphism counts:

$$|\mathsf{IndSub}(\Phi, k \to G)| = \sum_{H'} a_{\Phi,k}(H') \cdot |\mathsf{Hom}(H' \to G)| \,. \tag{3}$$

Note that the sum is over all possible graphs H'. Now, the algebraic approach to hardness states that $a_{\Phi,k}(H') \neq 0$ whenever H' is an edge-transitive graph with a prime-power number of edges, and Φ distinguishes H' from an independent set of size $|V(H')|$.[13] Therefore, if one can find some H', Φ that satisfy these requirements, one can apply again complexity monotonicity. We show that these requirements hold when H' is the subdivision of an (ℓ, ℓ)-biclique with ℓ a power of 2, and Φ is one of several interesting properties, including for example all non-trivial minor-closed properties of unbounded independence number. Then we apply complexity monotonicity; since the class of all bicliques has unbounded treewidth, and a subdivision of a biclique has the biclique as an F-gadget, we obtain hardness of $\#\textsc{IndSub}_{\mathrm{D}}(\Phi)$ from our results on $\#\textsc{Hom}_{\mathrm{D}}(C)$, yielding Theorems 7 and 8.

[13]More precisely, the algebraic approach applies to an intermediate, vertex-coloured version of the problem. We will carefully introduce the vertex-coloured version when needed, and show that it is interreducible with the uncoloured version, even in case of degenerate graphs G.

VI. Open Questions

The first question that we leave open is finding a complexity dichotomy into FPT versus $\#\mathsf{W}[1]$-hard cases for the problem of counting homomorphisms in degenerate graphs.

Open Problem 1. *Find an explicit criterion on computable graph classes C such that $\#\textsc{Hom}_{\mathrm{D}}(C)$ is FPT if the criterion is satisfied and $\#\mathsf{W}[1]$-hard otherwise.*

While dichotomies of this kind are not always possible [49], we believe that in our case a dichotomy exists. If this happens to be true, our work shows that the relevant parameter of C must lie between dag treewidth (which, if bounded, induces fixed-parameter tractability) and induced grid minors size (which, if unbounded, induces $\#\mathsf{W}[1]$-hardness). In fact, it might be the case that dag treewidth and induced grid minor size are equivalent, in the sense that C has bounded dag treewidth if and only if it has bounded induced grid minors. This would be a natural parallel of the Excluded Grid Theorem of [45], which says that C has bounded treewidth if and only if it has bounded (not necessarily induced) grid minors. However, the dag treewidth of a graph H is related to the acyclic orientations of its edges, which seem a priori unrelated to the size of the induced grid minors. Therefore, if an equivalent of the Excluded Grid Theorem holds for dag treewidth and induced grid minors, we expect its proof to require novel structural insights.

A second open question concerns the hardness of approximate counting in degenerate graphs. As mentioned earlier, complexity dichotomies for approximate counting problems are believed to be not always possible [25], and even for not necessarily degenerate host graphs, the complexity of approximating subgraph counts is only partially resolved [4]. Consequently, a complete picture of the complexity of approximate pattern counting in degenerate graphs seems elusive at this point. A good candidate for further research is the problem of counting induced k-matchings in degenerate graphs:

Open Problem 2. *Let k be a positive integer and let G be an n-vertex graph of degeneracy d. Is it possible to compute (with high probability) an ε-approximation of the number of induced k-matchings in G in time*

$$f(k,d) \cdot \mathsf{poly}(|V(G)|, \varepsilon^{-1})$$

for some computable function f?

To see why this is a good candidate, note that the exact counting version is hard (Theorem 5), and the decision version is fixed-parameter tractable [30]. Moreover, k-matchings are the minimal class of patterns for which our algorithm for approximating induced subgraph counts (Theorem 13) does not yield fixed-parameter tractability.

ACKNOWLEDGEMENTS

Marc Roth is grateful to Lior Gishboliner for introducing him to pattern counting in degenerate graphs. Marco Bressan was supported in part by the Google Focused Award "Algorithms and Learning for AI" (ALL4AI).

REFERENCES

[1] A. Abboud and V. V. Williams, "Popular conjectures imply strong lower bounds for dynamic problems," in *Proc. of IEEE FOCS*, 2014, pp. 434–443.

[2] N. Alon, P. Dao, I. Hajirasouliha, F. Hormozdiari, and S. C. Sahinalp, "Biomolecular network motif counting and discovery by color coding," *Bioinformatics*, vol. 24, no. 13, pp. i241–i249, 07 2008.

[3] M. Arenas, L. A. Croquevielle, R. Jayaram, and C. Riveros, "When is approximate counting for conjunctive queries tractable?" in *Proc. of ACM STOC*, 2021, p. 1015–1027.

[4] V. Arvind and V. Raman, "Approximation Algorithms for Some Parameterized Counting Problems," in *Proc. of ISAAC*, 2002, pp. 453–464.

[5] S. K. Bera, N. Pashanasangi, and C. Seshadhri, "Linear Time Subgraph Counting, Graph Degeneracy, and the Chasm at Size Six," in *Proc. of ITCS*, vol. 151, 2020, pp. 38:1–38:20. [Online]. Available: https://drops.dagstuhl.de/opus/volltexte/2020/11723

[6] ——, "Near-linear time homomorphism counting in bounded degeneracy graphs: The barrier of long induced cycles," in *Proc. of ACM-SIAM SODA*, 2021, pp. 2315–2332.

[7] M. Bressan, "Faster Subgraph Counting in Sparse Graphs," in *Proc. of IPEC*, vol. 148, 2019, pp. 6:1–6:15. [Online]. Available: https://drops.dagstuhl.de/opus/volltexte/2019/11467

[8] ——, "Faster algorithms for counting subgraphs in sparse graphs," *Algorithmica*, 2021. [Online]. Available: https://doi.org/10.1007/s00453-021-00811-0

[9] M. Bressan and M. Roth, "Exact and approximate pattern counting in degenerate graphs: New algorithms, hardness results, and complexity dichotomies," *CoRR*, vol. abs/2103.05588, 2021. [Online]. Available: https://arxiv.org/abs/2103.05588

[10] K. Bringmann and J. Slusallek, "Current algorithms for detecting subgraphs of bounded treewidth are probably optimal," in *Proc. of ICALP*, vol. 198, 2021, pp. 40:1–40:16.

[11] A. A. Bulatov and S. Zivný, "Approximate Counting CSP Seen from the Other Side," *ACM Trans. Comput. Theory*, vol. 12, no. 2, pp. 11:1–11:19, 2020.

[12] H. Chen and S. Mengel, "Counting Answers to Existential Positive Queries: A Complexity Classification," in *Proc. of ACM PODS*, 2016, pp. 315–326.

[13] J. Chen, B. Chor, M. Fellows, X. Huang, D. W. Juedes, I. A. Kanj, and G. Xia, "Tight lower bounds for certain parameterized NP-hard problems," *Inf. Comput.*, vol. 201, no. 2, pp. 216–231, 2005.

[14] J. Chen, X. Huang, I. A. Kanj, and G. Xia, "Strong computational lower bounds via parameterized complexity," *J. Comput. Syst. Sci.*, vol. 72, no. 8, pp. 1346–1367, 2006.

[15] Y. Chen, M. Thurley, and M. Weyer, "Understanding the Complexity of Induced Subgraph Isomorphisms," in *Proc. of ICALP*, 2008, pp. 587–596.

[16] N. Chiba and T. Nishizeki, "Arboricity and subgraph listing algorithms," *SIAM J. Comput.*, vol. 14, no. 1, pp. 210–223, 1985. [Online]. Available: http://dx.doi.org/10.1137/0214017

[17] R. Curticapean, "Counting matchings of size k is #W[1]-hard," in *Proc. of ICALP*, vol. 7965, 2013, pp. 352–363.

[18] ——, "The simple, little and slow things count: On parameterized counting complexity," Ph.D. dissertation, Saarland University, 2015. [Online]. Available: http://scidok.sulb.uni-saarland.de/volltexte/2015/6217/

[19] R. Curticapean, H. Dell, and D. Marx, "Homomorphisms are a good basis for counting small subgraphs," in *Proc. of ACM STOC*, 2017, pp. 210–223.

[20] R. Curticapean and D. Marx, "Complexity of Counting Subgraphs: Only the Boundedness of the Vertex-Cover Number Counts," in *Proc. of IEEE FOCS*, 2014, pp. 130–139.

[21] V. Dalmau and P. Jonsson, "The complexity of counting homomorphisms seen from the other side," *Theor. Comput. Sci.*, vol. 329, no. 1-3, pp. 315–323, 2004.

[22] H. Dell, J. Lapinskas, and K. Meeks, "Approximately counting and sampling small witnesses using a colourful decision oracle," in *Proc. of ACM-SIAM SODA*, 2020, pp. 2201–2211.

[23] J. Dörfler, M. Roth, J. Schmitt, and P. Wellnitz, "Counting Induced Subgraphs: An Algebraic Approach to #W[1]-hardness," in *Proc. of MFCS*, vol. 138, 2019, pp. 26:1–26:14.

[24] A. Durand and S. Mengel, "Structural Tractability of Counting of Solutions to Conjunctive Queries," *Theory Comput. Syst.*, vol. 57, no. 4, pp. 1202–1249, 2015.

[25] M. E. Dyer, L. A. Goldberg, and M. Jerrum, "An approximation trichotomy for Boolean #CSP," *J. Comput. Syst. Sci.*, vol. 76, no. 3-4, pp. 267–277, 2010.

[26] T. Eden, D. Ron, and W. Rosenbaum, "Almost optimal bounds for sublinear-time sampling of k-cliques: Sampling cliques is harder than counting," *CoRR*, vol. abs/2012.04090, 2020.

[27] T. Eden, D. Ron, and C. Seshadhri, "Faster sublinear approximation of the number of k-cliques in low-arboricity graphs," in *Proc. of ACM-SIAM SODA*, 2020, pp. 1467–1478.

[28] D. Eppstein, "Subgraph Isomorphism in Planar Graphs and Related Problems," *J. Graph Algorithms Appl.*, vol. 3, no. 3, pp. 1–27, 1999.

[29] D. Eppstein, M. Löffler, and D. Strash, "Listing all maximal cliques in large sparse real-world graphs," *ACM J. Exp. Algorithmics*, vol. 18, Nov. 2013.

[30] R. Erman, L. Kowalik, M. Krnc, and T. Walen, "Improved induced matchings in sparse graphs," *Discret. Appl. Math.*, vol. 158, no. 18, pp. 1994–2003, 2010.

[31] J. Flum and M. Grohe, "The Parameterized Complexity of Counting Problems," *SIAM J. Comput.*, vol. 33, no. 4, pp. 892–922, 2004.

[32] L. Gishboliner, Y. Levanzov, and A. Shapira, "Counting subgraphs in degenerate graphs," *Electron. Colloquium Comput. Complex.*, vol. 27, p. 177, 2020. [Online]. Available: https://eccc.weizmann.ac.il/report/2020/177

[33] L. Gishboliner, Y. Levanzov, A. Shapira, and R. Yuster, "Counting homomorphic cycles in degenerate graphs," *CoRR*, vol. abs/2011.05957, 2020. [Online]. Available: https://arxiv.org/abs/2011.05957

[34] M. Grohe, "The complexity of homomorphism and constraint satisfaction problems seen from the other side," *J. ACM*, vol. 54, no. 1, pp. 1:1–1:24, 2007.

[35] R. Impagliazzo and R. Paturi, "On the Complexity of k-SAT," *J. Comput. Syst. Sci.*, vol. 62, no. 2, pp. 367–375, 2001.

[36] M. Jerrum and K. Meeks, "The parameterised complexity of counting connected subgraphs and graph motifs," *J. Comput. Syst. Sci.*, vol. 81, no. 4, pp. 702–716, 2015.

[37] P. W. Kasteleyn, "Dimer Statistics and Phase Transitions," *J. Math. Phys.*, vol. 4, no. 2, pp. 287–293, 1963.

[38] B. Komarath, A. Pandey, and C. S. Rahul, "Graph homomorphism polynomials: Algorithms and complexity," *CoRR*, vol. abs/2011.04778, 2020. [Online]. Available: https://arxiv.org/abs/2011.04778

[39] R. Li, Q. Song, X. Xiao, L. Qin, G. Wang, J. Yu, and R. Mao, "I/O-efficient algorithms for degeneracy computation on massive networks," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2020.

[40] L. Lovász, *Large Networks and Graph Limits*, ser. Colloquium Publications. American Mathematical Society, 2012, vol. 60.

[41] D. Marx, "Can You Beat Treewidth?" *Theory Comput.*, vol. 6, no. 1, pp. 85–112, 2010.

[42] ——, "Tractable hypergraph properties for constraint satisfaction and conjunctive queries," *J. ACM*, vol. 60, no. 6, pp. 42:1–42:51, 2013.

[43] K. Meeks, "The challenges of unbounded treewidth in parameterised subgraph counting problems," *Discrete Appl. Math.*, vol. 198, pp. 170–194, 2016.

[44] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network Motifs: Simple Building Blocks of Complex Networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.

[45] N. Robertson and P. D. Seymour, "Graph minors. V. Excluding a planar graph," *J. Comb. Theory, Ser. B*, vol. 41, no. 1, pp. 92–114, 1986.

[46] M. Roth, "Counting Problems on Quantum Graphs: Parameterized and Exact Complexity Classifications," Ph.D. dissertation, Saarland University, 2019. [Online]. Available: https://scidok.sulb.uni-saarland.de/bitstream/20.500.11880/27575/1/thesis.pdf

[47] M. Roth and J. Schmitt, "Counting induced subgraphs: A Topological Approach to #W[1]-hardness," in *Proc. of IPEC*, 2018, pp. 24:1–24:14.

[48] M. Roth, J. Schmitt, and P. Wellnitz, "Counting small induced subgraphs satisfying monotone properties," in *Proc. of IEEE FOCS*, 2020, pp. 1356–1367.

[49] M. Roth and P. Wellnitz, "Counting and Finding Homomorphisms is Universal for Parameterized Complexity Theory," in *Proc. of ACM-SIAM SODA*, 2020, pp. 2161–2180.

[50] B. Schiller, S. Jager, K. Hamacher, and T. Strufe, "StreaM - A Stream-Based Algorithm for Counting Motifs in Dynamic Graphs," in *Algorithms for Computational Biology*, Cham, 2015, pp. 53–67.

[51] H. N. V. Temperley and M. E. Fisher, "Dimer problem in statistical mechanics – an exact result," *Philos. Mag.*, vol. 6, no. 68, pp. 1061–1063, 1961.