Counting Subgraphs in Somewhere Dense Graphs

Marco Bressan
Department of Computer Science, University of Milan, Italy

Leslie Ann Goldberg
Department of Computer Science, University of Oxford, UK

Kitty Meeks
School of Computing Science, University of Glasgow, UK

Marc Roth
Department of Computer Science, University of Oxford, UK

Abstract

We study the problems of counting copies and induced copies of a small pattern graph H in a large host graph G. Recent work fully classified the complexity of those problems according to structural restrictions on the patterns H. In this work, we address the more challenging task of analysing the complexity for restricted patterns *and* restricted hosts. Specifically we ask which families of allowed patterns and hosts imply fixed-parameter tractability, i.e., the existence of an algorithm running in time $f(H) \cdot |G|^{O(1)}$ for some computable function f. Our main results present exhaustive and explicit complexity classifications for families that satisfy natural closure properties. Among others, we identify the problems of counting small matchings and independent sets in subgraph-closed graph classes $\mathcal{G}$ as our central objects of study and establish the following crisp dichotomies as consequences of the Exponential Time Hypothesis:

- Counting k-matchings in a graph $G \in \mathcal{G}$ is fixed-parameter tractable if and only if $\mathcal{G}$ is nowhere dense.
- Counting k-independent sets in a graph $G \in \mathcal{G}$ is fixed-parameter tractable if and only if $\mathcal{G}$ is nowhere dense.

Moreover, we obtain almost tight conditional lower bounds if $\mathcal{G}$ is somewhere dense, i.e., not nowhere dense. These base cases of our classifications subsume a wide variety of previous results on the matching and independent set problem, such as counting k-matchings in bipartite graphs (Curticapean, Marx; FOCS 14), in F-colourable graphs (Roth, Wellnitz; SODA 20), and in degenerate graphs (Bressan, Roth; FOCS 21), as well as counting k-independent sets in bipartite graphs (Curticapean et al.; Algorithmica 19).

At the same time our proofs are much simpler: using structural characterisations of somewhere dense graphs, we show that a colourful version of a recent breakthrough technique for analysing pattern counting problems (Curticapean, Dell, Marx; STOC 17) applies to *any* subgraph-closed somewhere dense class of graphs, yielding a unified view of our current understanding of the complexity of subgraph counting.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Mathematics of computing → Graph theory

Keywords and phrases counting problems, somewhere dense graphs, parameterised complexity theory

Digital Object Identifier 10.4230/LIPIcs.ITCS.2023.27

Related Version *Full Version*: https://arxiv.org/abs/2209.03402

Funding This research was funded in whole, or in part, by EPSRC grant EP/V032305/1. For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. All data is provided in full in the results section of this paper.

1 Introduction

We study the following subgraph counting problem: given two graphs H and G, compute the number of copies of H in G. For several decades this problem has received widespread attention from the theoretical community, leading to a rich algorithmic toolbox that draws from different techniques [40, 2, 8, 30] and to deep structural results in parameterised complexity theory [20, 13]. Since it was discovered that subgraph counts reveal global properties of complex networks [36, 37], subgraph counting has also found several applications in fields such as biology [1, 44] genetics [46], phylogeny [31], and data mining [47]. Unfortunately, the subgraph counting problem is in general intractable, since it contains as special cases hard problems such as CLIQUE. This does not mean however that the problem is *always* intractable; it just means that it is tractable when the pattern H is restricted to certain graph families. Identifying these families of patterns that are efficiently countable has been a key question for the last twenty years. A long stream of research eventually showed that, unless standard conjectures fail, subgraph counting is tractable only for very restricted families of patterns [20, 17, 10, 15, 29, 35, 13, 42, 22].

To circumvent this "wall of intractability", in this work we restrict both the family of the pattern H and the family of the host G. Formally, given two classes of graphs $\mathcal{H}$ and $\mathcal{G}$, we study the problems $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$, $\#\textsc{IndSub}(\mathcal{H} \to \mathcal{G})$, and $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$, defined as follows. For all of them, the input is a pair (H, G) with $H \in \mathcal{H}$ and $G \in \mathcal{G}$. The outputs are respectively the number of subgraphs of G isomorphic to H, denoted by $\#\mathsf{Sub}(H \to G)$, the number of induced subgraphs of G isomorphic to H, denoted by $\#\mathsf{IndSub}(H \to G)$, and the number of homomorphisms (edge-preserving maps) from H to G, denoted by $\#\mathsf{Hom}(H \to G)$. Our goal is to determine for which $\mathcal{H}$ and $\mathcal{G}$ these three problems are tractable. To formalize what we mean by tractable, we adopt the framework of parameterized complexity [16]: we say that a problem is *fixed-parameter tractable*, or in the class FPT, if it is solvable in time $f(|H|) \cdot |G|^{O(1)}$ for some computable function f. For instance, we consider as tractable a running time of $2^{O(|H|)} \cdot |G|$ but not one of $|G|^{O(|H|)}$. This captures the intuition that H is "small" compared to G, and is the main theoretical framework for subgraph counting [20]. Thus, the goal of this work is understanding the fixed-parameter tractability of $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$, $\#\textsc{IndSub}(\mathcal{H} \to \mathcal{G})$, and $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$ as a function of $\mathcal{H}$ and $\mathcal{G}$. Moreover, when those problems are not fixed-parameter tractable we aim to show that they are hard for the complexity class #W[1], which can be thought of as the equivalent of NP for parameterized counting.

We first briefly discuss which properties of $\mathcal{G}$ are worthy of attention. When $\mathcal{G}$ is the class of all graphs, it is well known that each of the three problems is either FPT or #W[1]-hard depending on whether certain structural parameters of $\mathcal{H}$ (such as treewidth or vertex cover number) are bounded or not. Thus, when $\mathcal{G}$ is the class of all graphs, the problem is solved. However, when $\mathcal{G}$ is arbitrary, no such characterization is known. This is partly due to the fact that "natural" structural properties related to subgraph counting are harder to find for $\mathcal{G}$ than for $\mathcal{H}$; subgraph counting algorithms themselves usually exploit the structure of H but not that of G (think of tree decompositions). There is however one deep structural property that, if held by $\mathcal{G}$, yields tractability: the property of being *nowhere dense*, introduced by Nešetřil and Ossona de Mendez [38]. In a nutshell $\mathcal{G}$ is nowhere dense if, for all $r \in \mathbb{N}_0$, its members do not contain as subgraphs the r-subdivisions of arbitrarily large cliques; it can be shown that this generalizes several natural definitions of sparsity, including having bounded degree or bounded local treewidth, or excluding some topological minor. In a remarkable result, Nešetřil and Ossona de Mendez proved:

▶ **Theorem 1** (Theorem 18.9 in [39]). *If $\mathcal{G}$ is nowhere dense then the problems* $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$, $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$*, and* $\#\textsc{IndSub}(\mathcal{H} \to \mathcal{G})$ *are fixed-parameter tractable and can be solved in time* $f(|H|) \cdot |V(G)|^{1+o(1)}$ *for some computable function* f*.*

We note that, in the realm of decision problems, an even more general meta-theorem is known for first-order model-checking on nowhere dense graphs [27].

Thus the case that $\mathcal{G}$ is nowhere dense is closed, and we can focus on its complement – the case where $\mathcal{G}$ is *somewhere dense*. Hence the question studied in this work is: when are $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$, $\#\textsc{IndSub}(\mathcal{H} \to \mathcal{G})$, and $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$ fixed-parameter tractable, provided that $\mathcal{G}$ is somewhere dense?

2 Our Results

We prove dichotomies for $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$, $\#\textsc{IndSub}(\mathcal{H} \to \mathcal{G})$, and $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$ into FPT and #W[1]-hard cases, assuming that $\mathcal{G}$ is somewhere dense. It is known [43] that a fully general dichotomy is impossible even assuming that $\mathcal{G}$ is somewhere dense; thus we focus on the natural cases where $\mathcal{H}$ and/or $\mathcal{G}$ are monotone (closed under taking subgraphs) or hereditary (closed under taking induced subgraphs). Our dichotomies are expressed in terms of the finiteness of combinatorial parameters of $\mathcal{H}$ and $\mathcal{G}$, such as their clique number or their induced matching number. Existing complexity dichotomies for subgraph counting are based on using interpolation to evaluate linear combinations of homomorphism counts [13]. This technique has been exploited for families of host graphs that are closed under tensoring – the closure is used to create new instances for the interpolation. The host graphs in our dichotomy theorems do not have this closure property. Nevertheless, we obtain a dichotomy for all somewhere dense classes using a combination of techniques involving graph fractures and colourings.

The rest of this extended abstract presents our main conceptual contribution (Section 2.1), gives a detailed walk-through of our complexity dichotomies (Sections 2.2 - 2.4), provides some context (Section 3), and overviews the techniques behind our proofs (Section 4). For full proofs of our claims we refer the reader to the full version of this paper.

Basic preliminaries

We concisely state some necessary definitions and observations. We denote by $\mathcal{U}$ the class of all graphs. We denote by $\omega(G), \alpha(G), \beta(G)$, and $\mathrm{m}(G)$ respectively the clique, independence, biclique, and matching number of a graph G. The notation extends to graph classes by taking the supremum over their elements. Induced versions of those quantities are identified by the subscript $_{\mathsf{ind}}$ (for instance, $\mathrm{m}_{\mathsf{ind}}$ denotes the induced matching number). G^r denotes the r-subdivision of G, and $F \times G$ denotes the tensor product of F and G. All of our lower bounds assume the Exponential Time Hypothesis (ETH) [28]; and most of them rule out algorithms running in time $f(k) \cdot n^{o(k/\log k)}$ for any function f, and are therefore tight except possibly for a $O(\log k)$ factor in the exponent.[1] All of our #W[1]-hardness results are actually #W[1]-completeness results; this holds because $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$, $\#\textsc{IndSub}(\mathcal{H} \to \mathcal{G})$, and $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$ are always in #W[1] due to a characterisation of #W[1] via parameterised model-counting problems (see [21, Chapter 14]).

[1] This $O(\log k)$ gap is not an artifact of our proofs, but a consequence of the well-known open problem "Can you beat treewidth?" [33, 34].

2.1 Simpler Hardness Proofs for More Graph Families

Our first and most conceptual contribution is a novel approach to proving hardness of parameterized subgraph counting problems for somewhere-dense families of host graphs. This approach allows us to significantly generalize existing results while simultaneously yielding surprisingly simpler proofs.

The starting point is the observation that proving intractability results for parameterized counting problems is discouragingly difficult, as it often requires tedious and involved arguments. For instance, after Flum and Grohe conjectured that counting k-matchings is #W[1]-hard [20], the first proof required nine years and relied on sophisticated algebraic techniques [11]. This partially changed in 2017 when Curticapean, Dell and Marx [13] showed how to express a subgraph count $\#\mathsf{Sub}(H \to G)$ as linear combination of homomorphism counts $\sum_F a_F \cdot \#\mathsf{Hom}(F \to G)$, and that computing that linear combination has the same complexity as computing the hardest term $\#\mathsf{Hom}(F \to G)$ such that $a_F \neq 0$; a similar claim holds for induced subgraph counts as well. Thanks to this technique one can prove intractability of several subgraph counting problems, including for instance the problem of counting k-matchings.[2] These hardness results ultimately yielded complexity dichotomies for general subgraph counting problems, including notably $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$ and $\#\textsc{IndSub}(\mathcal{H} \to \mathcal{G})$ when $\mathcal{G}$ is the class of all graphs.

The technique of [13] does not work for proving hardness of $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$ and $\#\textsc{IndSub}(\mathcal{H} \to \mathcal{G})$ when $\mathcal{G} \neq \mathcal{U}$. Indeed, one caveat of that technique is that the family of host graphs $\mathcal{G}$ must satisfy certain conditions. One of those conditions is that $\mathcal{G}$ is closed under tensoring, i.e., that $G \times G' \in \mathcal{G}$ for all $G \in \mathcal{G}$ and all $G' \in \mathcal{U}$. The reason is that the interpolation relies on evaluating, say, $\mathsf{Sub}(H \to G \times G_i)$ for several carefully chosen graphs G_i, with the goal of constructing a certain invertible system of linear equations; for this to yield a reduction towards counting patterns in graphs from $\mathcal{G}$, it is crucial that $G \times G_i \in \mathcal{G}$ for all such G_i (Section 4 gives a concrete example using the problem of counting k-matchings). This is why the technique of [13] works smoothly for $\mathcal{G} = \mathcal{U}$; closure under tensoring holds trivially in that case. But many other natural graph families $\mathcal{G}$ are not closed under tensoring, including somewhere dense ones (for instance, the family of d-degenerate graphs for any fixed integer $d \geq 2$). Until now, this has been the main obstacle towards proving hardness of subgraph counting for arbitrary somewhere dense graph families. The central insight of our work is that this obstacle can be circumvented in a surprisingly simple way. Using well-established results from the theory of sparsity, we prove the following claim, which we explain in detail in Section 4:

Every monotone and somewhere dense class of graphs is closed under vertex-colourful tensor products of subdivided graphs.

Ignoring for a moment its technicalities, this result allows us to lift the interpolation technique via graph tensors to *any* monotone somewhere dense class of host graphs, including for instance the aforementioned class of d-degenerate graphs. In turn this yields complexity classifications for $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$, $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$, and $\#\textsc{IndSub}(\mathcal{H} \to \mathcal{G})$ that subsume and significantly strengthen almost all classifications known in the literature (see below). Moreover, our approach yields simple and almost self-contained proofs, helping understand the underlying causes of the hardness.

[2] In the field of database theory a similar technique expressing answers to unions of conjunctive queries as linear combinations of answers of conjunctive queries was independently discovered by Chen and Mengel [9].

2.2 The Complexity of $\#\mathrm{Sub}(\mathcal{H} \to \mathcal{G})$

This section presents our results on the fixed-parameter tractability of $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$. We start by presenting a minimal[3] family $\mathcal{H}$ for which hardness holds: the family of all k-matchings (or 1-regular graphs). In this case we also denote $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$ as $\#\textsc{Match}(\mathcal{G})$. In the foundational work by Flum and Grohe [20], $\#\textsc{Match}(\mathcal{U})$ was identified as a central problem because of the significance of its classical counterpart (counting the number of perfect matchings); a series of works then identified $\#\textsc{Match}(\mathcal{U})$ as the minimal intractable case [11, 15, 13]. In this work, we show that $\#\textsc{Match}(\mathcal{G})$ is the minimal hard case for *every* class $\mathcal{G}$ that is monotone and somewhere dense:

▶ **Theorem 2.** *Let $\mathcal{G}$ be a monotone class of graphs[4] and assume that ETH holds. Then $\#\textsc{Match}(\mathcal{G})$ is fixed-parameter tractable if and only if $\mathcal{G}$ is nowhere dense. More precisely, if $\mathcal{G}$ is nowhere dense then $\#\textsc{Match}(\mathcal{G})$ can be solved in time $f(k) \cdot |V(G)|^{1+o(1)}$ for some computable function f; otherwise $\#\textsc{Match}(\mathcal{G})$ is $\#\mathrm{W}[1]$-hard and cannot be solved in time $f(k) \cdot |G|^{o(k/\log k)}$ for any function f.*

Theorem 2 subsumes the existing intractability results for counting k-matchings in bipartite graphs [15], in F-colourable graphs [43], in bipartite graphs with one-sided degree bounds [14], and in degenerate graphs [7]. It also strengthens the latter result: while [7] establishes hardness of counting k-matchings in ℓ-degenerate graphs for $k + \ell$ as a parameter, Theorem 2 yields hardness for d-degenerate graphs for every fixed $d \geq 2$.[5] Additionally, we show that Theorem 2 cannot be strengthened to achieve polynomial-time tractability of $\#\textsc{Match}(\mathcal{G})$ for nowhere dense and monotone $\mathcal{G}$, unless $\#\mathrm{P} = \mathrm{P}$.

As a consequence of Theorem 2 we obtain, for hereditary $\mathcal{H}$, an exhaustive and detailed classification of the complexity of $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$ as a function of invariants of $\mathcal{G}$ and $\mathcal{H}$.

▶ **Theorem 3.** *Let $\mathcal{H}$ and $\mathcal{G}$ be graph classes such that $\mathcal{H}$ is hereditary and $\mathcal{G}$ is monotone. Then the complexity of $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$ is exhaustively classified by Table 1.*

Table 1 The complexity of $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$ for hereditary $\mathcal{H}$ and monotone $\mathcal{G}$. Here "hard" means $\#\mathrm{W}[1]$-hard and, unless ETH fails, without an algorithm running in time $f(|H|) \cdot |G|^{o(|V(H)|/\log |V(H)|)}$; "hard†" means the same, but without an algorithm running in $f(|H|) \cdot |G|^{o(|V(H)|)}$.

	$\mathcal{G}$ n. dense	$\mathcal{G}$ s. dense $\omega(\mathcal{G}) = \infty$	$\mathcal{G}$ s. dense $\omega(\mathcal{G}) < \infty$ $\beta(\mathcal{G}) = \infty$	$\mathcal{G}$ s. dense $\omega(\mathcal{G}) < \infty$ $\beta(\mathcal{G}) < \infty$
$\mathrm{m}(\mathcal{H}) < \infty$	P	P	P	P
$\mathrm{m}_{\mathsf{ind}}(\mathcal{H}) = \infty$	FPT	hard	hard	hard
$\mathrm{m}_{\mathsf{ind}}(\mathcal{H}) < \infty$, $\beta_{\mathsf{ind}}(\mathcal{H}) = \infty$	P	hard†	hard†	P
Otherwise	P	hard†	P	P

[3] Minimal means that, for every class $\mathcal{H}'$, $\#\textsc{Sub}(\mathcal{H}' \to \mathcal{G})$ is intractable if and only if the monotone closure of $\mathcal{H}'$ includes $\mathcal{H}$. The same holds for $\#\textsc{IndSub}$ with "monotone" replaced by "hereditary".

[4] We emphasize that we do not need our classes to be computable or recursively enumerable. This is due to the assumed closure properties of the classes.

[5] The class of all d-degenerate graphs is somewhere dense for all $d \geq 2$.

Note that the unique fixed-parameter tractability result in Table 1 is a "real" FPT case: we can show that, unless P = #P, it is in FPT but not in P.

From the classification of Theorem 3 one can derive interesting corollaries. For example, when $\mathcal{H}$ and $\mathcal{G}$ are monotone one has essentially the same classification of the case $\mathcal{G} = \mathcal{U}$: only the boundedness of the matching number of $\mathcal{H}$ (or equivalently, of its vertex-cover number) counts [15].

▶ **Theorem 4.** *Let $\mathcal{H}$ and $\mathcal{G}$ be monotone classes of graphs and assume that ETH holds. Then $\#\text{SUB}(\mathcal{H} \to \mathcal{G})$ is fixed-parameter tractable if $\mathrm{m}(\mathcal{H}) < \infty$ or $\mathcal{G}$ is nowhere dense; otherwise $\#\text{SUB}(\mathcal{H} \to \mathcal{G})$ is $\#\mathrm{W}[1]$-complete and cannot be solved in time $f(|H|) \cdot |G|^{o(|V(H)|/\log(|V(H)|))}$ for any function f.*

We conclude by remarking that Table 1 and the proofs of its bounds suggest the existence of three general algorithmic strategies for subgraph counting:

1. If $\mathcal{G}$ is nowhere dense (first column of Table 1), then one can use the FPT algorithm of Theorem 1, based on Gaifman's locality theorem for first-order formulas and the local sparsity of nowhere dense graphs (see [39]).
2. If $\mathrm{m}(\mathcal{H}) < \infty$ (first row of Table 1), then one can use the polynomial-time algorithm of Curticapean and Marx [15], based on guessing the image of a maximum matching of H and counting its extensions via dynamic programming.
3. All remaining entries marked as "P" are shown to be essentially trivial. Concretely, we will rely on Ramsey's theorem to prove that minor modifications of the naive brute-force approach yield polynomial-time algorithms for those cases.

2.3 The Complexity of $\#\text{IndSub}(\mathcal{H} \to \mathcal{G})$

In the previous section we proved that, when $\mathcal{G}$ is somewhere dense, k-matchings are the minimal hard family of patterns for $\#\text{SUB}(\mathcal{H} \to \mathcal{G})$. In this section we show that k-independent sets play a similar role for $\#\text{INDSUB}(\mathcal{H} \to \mathcal{G})$. Let $\#\text{INDSET}(\mathcal{G}) = \#\text{INDSUB}(\mathcal{I} \to \mathcal{G})$ where $\mathcal{I}$ is the set of all all independent sets (or 0-regular graphs). We prove:

▶ **Theorem 5.** *Let $\mathcal{G}$ be a monotone class of graphs and assume that ETH holds. Then $\#\text{INDSET}(\mathcal{G})$ is fixed-parameter tractable if and only if $\mathcal{G}$ is nowhere dense. More precisely, if $\mathcal{G}$ is nowhere dense then $\#\text{INDSET}(\mathcal{G})$ can be solved in time $f(k) \cdot |V(G)|^{1+o(1)}$ for some computable function f; otherwise $\#\text{INDSET}(\mathcal{G})$ cannot be solved in time $f(k) \cdot |G|^{o(k/\log k)}$ for any function f.*

This result subsumes the intractability result for counting k-independent sets in bipartite graphs of [12]. It also strengthens the result of [7], which shows $\#\text{INDSET}(\mathcal{G})$ is hard when parameterized by $k+d$ where d is the degeneracy of G. More precisely, [7] does not imply that $\#\text{INDSET}(\mathcal{G})$ is hard for $\mathcal{G}$ being the class of d-degenerate graphs, for any $d \geq 2$; Theorem 5 instead proves such hardness for *every* $d \geq 2$. Finally, we point out that the FPT case of Theorem 5 is not in P unless P = #P.

As consequence of Theorem 5, when $\mathcal{H}$ is hereditary (and thus in particular monotone) we obtain:

▶ **Theorem 6.** *Let $\mathcal{H}$ and $\mathcal{G}$ be classes of graphs such that $\mathcal{H}$ is hereditary and $\mathcal{G}$ is monotone. Then the complexity of $\#\text{INDSUB}(\mathcal{H} \to \mathcal{G})$ is exhaustively classified by Table 2.*

Table 2 The complexity of #IndSub($\mathcal{H} \to \mathcal{G}$) for hereditary $\mathcal{H}$ and monotone $\mathcal{G}$. Here "hard" means #W[1]-hard and, unless ETH fails, without an algorithm running in time $f(|H|) \cdot |G|^{o(|V(H)|/\log|V(H)|)}$; "hard†" means the same, but without an algorithm running in $f(|H|) \cdot |G|^{o(|V(H)|)}$.

	$\mathcal{G}$ n. dense	$\mathcal{G}$ s. dense $\omega(\mathcal{G}) = \infty$	$\mathcal{G}$ s. dense $\omega(\mathcal{G}) < \infty$ $\alpha(\mathcal{G}) = \infty$
$\lvert\mathcal{H}\rvert < \infty$	P	P	P
$\alpha(\mathcal{H}) = \infty$	FPT	hard†	hard
Otherwise	P	hard†	P

2.4 The Complexity of #Hom($\mathcal{H} \to \mathcal{G}$)

Finally, we study the parameterized complexity of #Hom($\mathcal{H} \to \mathcal{G}$). We denote by $\mathsf{tw}(H)$ the treewidth of a graph H. Informally, graphs of small treewidth admit a decomposition with small separators, which allows for efficient dynamic programming. In this work we use treewidth in a purely black-box fashion (e.g. via excluded-grid theorems); for its formal definition see [16, Chapter 7]. We prove:

▶ **Theorem 7.** *Let $\mathcal{H}$ and $\mathcal{G}$ be monotone classes of graphs.*

1. *If $\mathcal{G}$ is nowhere dense then #Hom($\mathcal{H} \to \mathcal{G}$) is fixed-parameter tractable and can be solved in time $f(|H|) \cdot |V(G)|^{1+o(1)}$ for some computable function f.*
2. *If $\mathsf{tw}(\mathcal{H}) < \infty$ then #Hom($\mathcal{H} \to \mathcal{G}$) is solvable in polynomial time, and if a tree decomposition of H of width t is given, then it can be solved in time $|H|^{O(1)} \cdot |V(G)|^{t+1}$.*
3. *If $\mathcal{G}$ is somewhere dense and $\mathsf{tw}(\mathcal{H}) = \infty$ then #Hom($\mathcal{H} \to \mathcal{G}$) is #W[1]-hard and, assuming ETH, cannot be solved in time $f(|H|) \cdot |G|^{o(\mathsf{tw}(H))}$ for any function f.*

(The novel part is 3.; we included 1. and 2. to provide the complete picture.)

Unfortunately, in contrast to #Sub and #IndSub, we do not know how to extend Theorem 7 to hereditary $\mathcal{H}$. We point out however that for hereditary $\mathcal{H}$ the finiteness of $\mathsf{tw}(\mathcal{H})$ cannot be the correct criterion: if $\mathcal{H}$ is the set of all complete graphs and $\mathcal{G}$ is the set of all bipartite graphs, then $\mathcal{H}$ is hereditary and $\mathsf{tw}(\mathcal{H}) = \infty$, but #Hom($\mathcal{H} \to \mathcal{G}$) is easy since $|V(H)| \leq 2$ or $\#\mathsf{Hom}(H \to G) = 0$. More generally, the complexity of #Hom($\mathcal{H} \to \mathcal{G}$) appears to be far from completely understood for arbitrary classes $\mathcal{H}$. In fact, it has been recently posed as an open problem even for specific monotone and somewhere dense $\mathcal{G}$ such as the family of d-degenerate graphs [7, 3]. There is some evidence that the finiteness of induced grid minors is the right criterion for tractability [7].

In what follows we provide a detailed exposition of our proof techniques, starting with a brief summary of the state of the art.

3 Related Work

The general idea of using interpolation as a reduction technique for counting problems dates back to the foundational work of Valiant [48]. Roughly speaking, the key to interpolation is constructing a system of linear equations that is invertible and thus has a unique solution. For example, in the classic case of polynomial interpolation (where one has to infer the coefficients of a univariate polynomial given an oracle that evaluates it) the system corresponds to a Vandermonde matrix, which is nonsingular and thus invertible. In the case of linear combinations of homomorphism counts, an invertible system of linear equations can be constructed via graph tensoring arguments, as proven implicitly by works of Lovász (see e.g. [32, Chapters 5 and 6]). It was then discovered by Curticapean et al. in [13] that these interpolation arguments could be extended to subgraph and induced subgraph counts, by showing that those counts may be expressed as linear combinations of homomorphism counts. Using this fact, they proved that interpolation through graph tensoring applies to a wide variety of parameterised subgraph counting problems. However, their technique fails when one restrict the class of host graphs $\mathcal{G}$, see above; our work shows how to circumvent this obstacle.

The idea of using graph subdivisions for proving hardness results appeared in the context of linear-time subgraph counting in degenerate graphs [4, 5, 3]. For example, [4] observed that counting triangles in general graphs, which is conjectured not to admit a linear time algorithm, reduces in linear time to counting 6-cycles in degenerate graphs by subdividing each edge once (which always yields a 2-degenerate graph). Our work makes heavy use of graph subdivisions as well, although in a more sophisticate fashion. This is not surprising since, for each $d \geq 2$, the class of d-degenerate graphs constitutes an example of a monotone somewhere dense class of graphs.

4 Overview of Our Techniques

The present section expands upon Section 2.1 and gives a detailed technical overview of our proofs of hardness for $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$ and $\#\textsc{IndSub}(\mathcal{H} \to \mathcal{G})$ (Section 4.1) and for $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$ (Section 4.2). Those hardness proofs contain virtually all the effort in characterizing the complexity of the problems, since the complexity of the tractable cases follows mostly from the running time of existing algorithms.

4.1 Classifying Subgraph and Induced Subgraph Counting

We start by analysing a simple case. Recall that a graph family $\mathcal{G}$ is somewhere dense if, for some $r \in \mathbb{N}_0$, for all $k \in \mathbb{N}$ there is a $G \in \mathcal{G}$ such that K_k^r is a subgraph of G. From this characterization it is immediate that, if $\mathcal{G}$ is somewhere dense *and monotone*, then it contains the r-subdivisions of every graph. In turn, this implies that detecting subdivisions of cliques in $\mathcal{G}$ is at least as hard as the parameterised clique problem [19]. Since the parameterised clique problem is W[1]-hard, we deduce that $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$ and $\#\textsc{IndSub}(\mathcal{H} \to \mathcal{G})$ are intractable when $\mathcal{H} = \{K_k^r : k, r \in \mathbb{N}\}$ and $\mathcal{G}$ is monotone and somewhere dense. Unfortunately, it is unclear how to extend this approach to arbitrary $\mathcal{H}$, since the elements of $\mathcal{H}$ are not necessarily r-subdivisions of graphs that are hard to count. To show how this obstacle can be overcome, we will focus on $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$ when $\mathcal{H}$ is the class of k-matchings, $\mathcal{M} = \{M_k : k \in \mathbb{N}\}$; in other words, on the problem of counting k-matchings, $\#\textsc{Sub}(\mathcal{M} \to \mathcal{G})$. This problem will turn out to be the minimal hard case for $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$, and its analysis will contain the key ingredients of our proof. The proof for $\#\textsc{IndSub}(\mathcal{H} \to \mathcal{G})$ will be similar.

Let us start by outlining the hardness proof of $\#\textsc{Sub}(\mathcal{M} \to \mathcal{G})$ when $\mathcal{G} = \mathcal{U}$, by using the interpolation technique discussed in Section 3. From [13], we know that for every $k \in \mathbb{N}$ there is a function $a_k : \mathcal{U} \to \mathbb{Q}$ with finite support such that, for every $G \in \mathcal{U}$,

$$\#\mathsf{Sub}(M_k \to G) = \sum_H a_k(H) \cdot \#\mathsf{Hom}(H \to G) \tag{1}$$

where the sum is over all isomorphism classes of all graphs. By a classic result of Dalmau and Jonsson [17], computing $\#\mathsf{Hom}(H \to G)$ is not fixed-parameter tractable for H of unbounded treewidth, unless ETH fails. Hence, if we could use (1) to show that an FPT algorithm for computing $\#\mathsf{Sub}(M_k \to G)$ yields an FPT algorithm for computing $\#\mathsf{Hom}(H \to G)$ for some H whose treewidth grows with k, we would conclude that computing $\#\mathsf{Sub}(M_k \to G)$ is not fixed-parameter tractable unless ETH fails. This is what [13] indeed prove. The idea is to apply (1) not to G, but to a set of carefully chosen graphs $\hat{G}_1, \ldots, \hat{G}_\ell$ such that the counts $\#\mathsf{Hom}(M_k \to \hat{G}_1), \ldots, \#\mathsf{Hom}(M_k \to \hat{G}_\ell)$ can be used to solve a linear system and infer $\#\mathsf{Hom}(H \to G)$ for all H appearing on the right-hand side of (1).

Let us explain this idea in more detail. Suppose we had an oracle for $\#\textsc{Sub}(\mathcal{M} \to \mathcal{U})$, so that we could quickly compute $\#\mathsf{Sub}(M_k \to G)$ for any desired G. Let ℓ be the size of the support of a_k, which as said above is finite and thus a function of k, and let $\{G_i\}_{i=1,\ldots,\ell}$ be a set of graphs such that each G_i has size bounded by a function of k. It is a well-known fact that, for all graphs H, G, G',

$$\#\mathsf{Hom}(H \to G \times G') = \#\mathsf{Hom}(H \to G) \cdot \#\mathsf{Hom}(H \to G'). \tag{2}$$

By combining (1) and (2), for each $i = 1, \ldots, \ell$ we obtain

$$\#\mathsf{Sub}(M_k \to G \times G_i) = \sum_H a_k(H) \cdot \#\mathsf{Hom}(H \to G_i) \cdot \#\mathsf{Hom}(H \to G) = \sum_{\substack{H \\ a_k(H) \neq 0}} b_H^i \cdot X_H \,, \tag{3}$$

where $b_H^i := \#\mathsf{Hom}(H \to G_i)$ and $X_H := a_k(H) \cdot \#\mathsf{Hom}(H \to G)$. Now, we can compute $\#\mathsf{Hom}(H \to G_i)$ in FPT time since $|G_i|$ is bounded by a function of k, and we can compute $\#\mathsf{Sub}(M_k \to G \times G_i)$ using the oracle. Therefore, in FPT time we can comput a system of ℓ linear equations with the X_H as unknowns. By applying classical results due to Lovász (see e.g. [32, Chapter 5]), Curticapean et al. [13] showed that there always exists a choice of the G_i's such that this system has a unique solution. Hence, using those G_i's one can compute $\#\mathsf{Hom}(H \to G)$ in FPT time for all H with $a_k(H) \neq 0$. In particular, one can compute $\#\mathsf{Hom}(F_k \to G)$ where F_k is any k-edge graph of maximal treewidth, since [13] also showed that $a_k(H) \neq 0$ for all H with $|E(H)| \leq k$. This gives a parameterized reduction from $\#\textsc{Hom}(\mathcal{F} \to \mathcal{U})$ to $\#\textsc{Sub}(\mathcal{M} \to \mathcal{U})$, where $\mathcal{F}$ is the class of all maximal-treewidth graphs F_k. Since $\#\textsc{Hom}(\mathcal{F} \to \mathcal{U})$ is hard by [17], the reduction establishes hardness of $\#\textsc{Sub}(\mathcal{M} \to \mathcal{U})$ as desired.

Our main question is whether this strategy can be extended from $\mathcal{U}$ to any monotone somewhere dense class $\mathcal{G}$. This it not obvious, since the argument above relies on two crucial ingredients that may be lost when moving from $\mathcal{U}$ to $\mathcal{G}$:

(I.1) We need to find a family of graphs $\hat{\mathcal{F}} = \{\hat{F}_k \mid k \in \mathbb{N}\}$ such that $\#\textsc{Hom}(\hat{\mathcal{F}} \to \mathcal{G})$ is hard and, for all $k \in \mathbb{N}$, $a_k(\hat{F}_k) \neq 0$.

(I.2) We need to find graphs G_i such that $G \times G_i \in \mathcal{G}$. This is necessary since the argument performs a reduction to the problem of counting $\#\mathsf{Sub}(M_k \to G \times G_i)$, and is not straightforward since $G \times G_i$ may not be in $\mathcal{G}$ even when both G, G_i are.

It turns out that both requirements can be satisfied in a systematic way. First, we study $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$ in some carefully chosen vertex-coloured and edge-coloured version. It is well-known that the coloured version of the problem is equivalent in complexity (in the FPT sense) to the uncoloured version; so, to make progress, we may consider the coloured version. Next, coloured graphs come with a canonical coloured version of the tensor product which satisfies (2), so we can hope to apply interpolation via tensor products in the colorful setting, too. The introduction of colours in the analysis of parameterised problems is a common tool for streamlining reductions that are otherwise unnecessarily complicated (see e.g. [15, 41, 18, 22]). The technical details of the coloured version are not hard, but cumbersome to state; since here we do not need them, we defer them to the full version. Let us now give a high-level explanation of how we achieve (I.1) and (I.2).

For (I.1), we let $\hat{\mathcal{F}}$ be the class of all r-subdivisions of a family $\mathcal{E}$ of regular expander graphs. A simple construction then allows us to reduce $\#\textsc{Hom}(\mathcal{E} \to \mathcal{U})$, which is known to be hard, to $\#\textsc{Hom}(\hat{\mathcal{F}} \to \mathcal{U}^r)$, where $\mathcal{U}^r$ is the set of all r-subdivisions of graphs. As noted above $\mathcal{U}^r \subseteq \mathcal{G}$, hence $\#\textsc{Hom}(\hat{\mathcal{F}} \to \mathcal{G})$ is hard. We show in the coloured version that for each graph $F_k \in \hat{\mathcal{F}}$ with k edges, $a_k(F_k) \neq 0$. Thus, (I.1) is satisfied.

For (I.2) we construct, for each k, a finite sequence of coloured graphs $G_1, G_2, \ldots$ satisfying the following two conditions: the system of linear equations given by (the coloured version of) (3) has a unique solution, and the coloured tensor product between each G_i and any coloured graph in $\mathcal{U}^r$ is in $\mathcal{G}$. Concretely, we choose as G_i the so-called *fractured graphs* of the r-subdivisions of the expanders in $\mathcal{E}$. Fractured graphs are obtained by a splitting operation on a graph and come with a natural vertex colouring. They have been introduced in recent work on classifying subgraph counting problems [41] and we describe them in detail in the full version.

Together, our resolutions of (I.1) and (I.2) yield a colourful version of the framework of [13] that applies to *any monotone somewhere dense class of host graphs.* As a consequence we obtain that $\#\textsc{Hom}(\mathcal{E} \to \mathcal{U})$, the problem of counting homomorphisms from expanders in $\mathcal{E}$ to arbitrary hosts graphs, reduces in FPT time to $\#\textsc{Sub}(\mathcal{M} \to \mathcal{G})$ whenever $\mathcal{G}$ is monotone and somewhere dense. Since $\#\textsc{Hom}(\mathcal{E} \to \mathcal{U})$ is intractable, this proves the hardness of $\#\textsc{Sub}(\mathcal{M} \to \mathcal{G})$ for all monotone and somewhere dense $\mathcal{G}$, as stated in Theorem 2. From this result we will then be able to prove our general classification for $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$ (Theorem 3) by combining existing results and Ramsey-type arguments on $\mathcal{H}$ and $\mathcal{G}$.

This concludes our overview for $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$. The proofs for $\#\textsc{IndSub}(\mathcal{H} \to \mathcal{G})$ are similar, but instead of $\#\textsc{Sub}(\mathcal{M} \to \mathcal{G})$, they use as a minimal hard case $\#\textsc{IndSet}(\mathcal{G})$, the problem of counting k-independent sets in host graphs from $\mathcal{G}$.

4.2 Classifying Homomorphism Counting via Wall Minors

The proof of our dichotomy for $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$ for monotone $\mathcal{H}$ and $\mathcal{G}$ (Theorem 7) requires us to establish hardness when $\mathcal{G}$ is somewhere dense and $\mathsf{tw}(\mathcal{H}) = \infty$. Recall that our solution of (I.1) relied on a reduction from (the coloured version of) $\#\textsc{Hom}(\mathcal{E} \to \mathcal{U})$ to (the coloured version of) $\#\textsc{Hom}(\hat{\mathcal{F}} \to \mathcal{U}^r)$, where $\mathcal{E}$ is a family of regular expander graphs, $\hat{\mathcal{F}}$ is the class of all r-subdivisions of graphs in $\mathcal{E}$, and $\mathcal{U}^r$ is the class of r-subdivisions of all graphs. Since for all monotone somewhere dense classes $\mathcal{G}$ there is an r such that $\mathcal{U}^r \subseteq \mathcal{G}$, we would be done if we could make sure that every monotone class of graphs of unbounded treewidth $\mathcal{H}$ contains $\hat{\mathcal{F}}$ as a subset. Unfortunately, this is not the case. As a trivial example, $\mathcal{H}$ could be the class of all graphs of degree at most 3 while $\mathcal{E}$ is a family of 4-regular expanders. Straightforward weakenings of this condition also fail.

To circumvent this problem, we use a result of Thomassen [45] to prove that, for every positive integer r and every monotone class of graphs $\mathcal{H}$ with unbounded treewidth, the following holds: for each *wall* $W_{k,k}$, the class $\mathcal{H}$ contains a subdivision of $W_{k,k}$ in which each edge is subdivided a positive multiple of r times. Now, the crucial property of the class of all walls $\mathcal{W} := \{W_{k,k} \mid k \in \mathbb{N}\}$ is that $\#\textsc{Hom}(\mathcal{W} \to \mathcal{U})$ is intractable by the classification of Dalmau and Jonsson [17]. Refining our constructions based on subdivided graphs, we are then able to show that $\#\textsc{Hom}(\mathcal{W} \to \mathcal{U})$ reduces to $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$ whenever $\mathcal{H}$ is monotone and of unbounded treewidth, and $\mathcal{G}$ is monotone and somewhere dense. Theorem 7 will then follow as a direct consequence.

5 Outlook

Due to the absence of a general dichotomy [43], the following two directions are evident candidates for future analysis.

Hereditary Host Graphs

Is there a way to refine our classifications to hereditary $\mathcal{G}$? While such results would naturally be much stronger, we point out that a classification of general first-order (FO) model-checking and model-counting in hereditary graphs is wide open. Concretely, even if $\mathcal{H} = \mathcal{U}$, it currently seems elusive to obtain criteria for hereditary $\mathcal{G}$ which, if satisfied, yield fixed-parameter tractability of $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$, $\#\textsc{IndSub}(\mathcal{H} \to \mathcal{G})$, and $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$ and which, if not satisfied, yield #W[1]-hardness of those problems. In a nutshell, the problem is that there are arbitrarily dense hereditary classes of host graphs for which those problems, and even the much more general FO-model counting problem, become tractable; a trivial example is given by $\mathcal{G}$ being the class of all complete graphs. See [23, 25, 26] for recent work on specific hereditary hosts and [24, 6] for general approaches to understand FO model checking on dense graphs.

Arbitrary Pattern Graphs

Can we refine our classifications to arbitrary classes of patterns $\mathcal{H}$, given that we stay in the realm of monotone classes of hosts $\mathcal{G}$? We believe this question is the most promising direction for future research. While a sufficient and necessary criterion for the fixed-parameter tractability of, say $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$, must depend on the set of forbidden subgraphs of $\mathcal{G}$, we conjecture that the structure of monotone somewhere dense graph classes is rich enough to allow for an explicit combinatorial description of such a criterion. In fact, such criteria have already been established for some specific classes of host graphs, e.g. bipartite graphs [15] and degenerate graphs [7].

References

1 Noga Alon, Phuong Dao, Iman Hajirasouliha, Fereydoun Hormozdiari, and S. Cenk Sahinalp. Biomolecular network motif counting and discovery by color coding. *Bioinformatics*, 24(13):i241–i249, 2008. doi:10.1093/bioinformatics/btn163.

2 Noga Alon, Raphael Yuster, and Uri Zwick. Color-Coding. *J. ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.

3 Suman K. Bera, Lior Gishboliner, Yevgeny Levanzov, C. Seshadhri, and Asaf Shapira. Counting subgraphs in degenerate graphs. *J. ACM*, 69(3):23:1–23:21, 2022. doi:10.1145/3520240.

4 Suman K. Bera, Noujan Pashanasangi, and C. Seshadhri. Linear time subgraph counting, graph degeneracy, and the chasm at size six. In *Proceedings of the 11th conference on Innovations in Theoretical Computer Science (ITCS)*, pages 38:1–38:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ITCS.2020.38.

5 Suman K. Bera, Noujan Pashanasangi, and C. Seshadhri. Near-linear time homomorphism counting in bounded degeneracy graphs: The barrier of long induced cycles. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2315–2332. SIAM, 2021. doi:10.1137/1.9781611976465.138.

6 Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width I: tractable FO model checking. *J. ACM*, 69(1):3:1–3:46, 2022. doi:10.1145/3486655.

7 Marco Bressan and Marc Roth. Exact and approximate pattern counting in degenerate graphs: New algorithms, hardness results, and complexity dichotomies. In *Proceedings of the 62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 276–285. IEEE, 2021. doi:10.1109/FOCS52979.2021.00036.

8 Russ Bubley and Martin E. Dyer. Path coupling: A technique for proving rapid mixing in markov chains. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 223–231. IEEE Computer Society, 1997. doi:10.1109/SFCS.1997.646111.

9 Hubie Chen and Stefan Mengel. Counting Answers to Existential Positive Queries: A Complexity Classification. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, pages 315–326. ACM, 2016. doi:10.1145/2902251.2902279.

10 Yijia Chen, Marc Thurley, and Mark Weyer. Understanding the Complexity of Induced Subgraph Isomorphisms. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 587–596. Springer, 2008. doi:10.1007/978-3-540-70575-8_48.

11 Radu Curticapean. Counting matchings of size k is W[1]-hard. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 352–363. Springer, 2013. doi:10.1007/978-3-642-39206-1_30.

12 Radu Curticapean, Holger Dell, Fedor V. Fomin, Leslie Ann Goldberg, and John Lapinskas. A fixed-parameter perspective on #BIS. *Algorithmica*, 81(10):3844–3864, 2019. doi:10.1007/s00453-019-00606-4.

13 Radu Curticapean, Holger Dell, and Dániel Marx. Homomorphisms are a good basis for counting small subgraphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 210–223. ACM, 2017. doi:10.1145/3055399.3055502.

14 Radu Curticapean, Holger Dell, and Marc Roth. Counting Edge-Injective Homomorphisms and Matchings on Restricted Graph Classes. In *Proceedings of the 34th Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 25:1–25:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.STACS.2017.25.

15 Radu Curticapean and Dániel Marx. Complexity of Counting Subgraphs: Only the Boundedness of the Vertex-Cover Number Counts. In *Proceedings of the 55th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 130–139. IEEE, 2014. doi:10.1109/FOCS.2014.22.

16 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

17 Víctor Dalmau and Peter Jonsson. The complexity of counting homomorphisms seen from the other side. *Theoret. Comput. Sci.*, 329(1-3):315–323, 2004. doi:10.1016/j.tcs.2004.08.008.

18 Julian Dörfler, Marc Roth, Johannes Schmitt, and Philip Wellnitz. Counting Induced Subgraphs: An Algebraic Approach to #W[1]-Hardness. *Algorithmica*, 84(2):379–404, 2022. doi:10.1007/s00453-021-00894-9.

19 Zdenek Dvorák, Daniel Král, and Robin Thomas. Testing first-order properties for subclasses of sparse graphs. *J. ACM*, 60(5):36:1–36:24, 2013. doi:10.1145/2499483.

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

21 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. doi:10.1007/3-540-29953-X.

22 Jacob Focke and Marc Roth. Counting small induced subgraphs with hereditary properties. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1543–1551. ACM, 2022. doi:10.1145/3519935.3520008.

23 Jakub Gajarský, Petr Hlinený, Daniel Lokshtanov, Jan Obdrzálek, Sebastian Ordyniak, M. S. Ramanujan, and Saket Saurabh. FO model checking on posets of bounded width. In *Proceedings of the 56th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 963–974. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.63.

24 Jakub Gajarský, Petr Hlinený, Jan Obdrzálek, Daniel Lokshtanov, and M. S. Ramanujan. A new perspective on FO model checking of dense graph classes. *ACM Trans. Comput. Log.*, 21(4):28:1–28:23, 2020. doi:10.1145/3383206.

25 Robert Ganian, Petr Hlinený, Daniel Král, Jan Obdrzálek, Jarett Schwartz, and Jakub Teska. FO model checking of interval graphs. *Log. Methods Comput. Sci.*, 11(4), 2015. doi:10.2168/LMCS-11(4:11)2015.

26 Colin Geniet and Stéphan Thomassé. First order logic and twin-width in tournaments and dense oriented graphs. *CoRR*, abs/2207.07683, 2022. arXiv:2207.07683, doi:10.48550/arXiv.2207.07683.

27 Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *J. ACM*, 64(3):17:1–17:32, 2017. doi:10.1145/3051095.

28 Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k-SAT. *J. Comput. System Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

29 Mark Jerrum and Kitty Meeks. The parameterised complexity of counting connected subgraphs and graph motifs. *J. Comput. System Sci.*, 81(4):702–716, 2015. doi:10.1016/j.jcss.2014.11.015.

30 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. doi:10.1145/1008731.1008738.

31 Oleksii Kuchaiev, Tijana Milenković, Vesna Memišević, Wayne Hayes, and Nataša Pržulj. Topological network alignment uncovers biological function and phylogeny. *Journal of the Royal Society Interface*, 7(50):1341–1354, 2010. doi:10.1098/rsif.2010.0063.

32 László Lovász. *Large Networks and Graph Limits*, volume 60 of *Colloquium Publications*. American Mathematical Society, 2012. URL: http://www.ams.org/bookstore-getitem/item=COLL-60.

33 Dániel Marx. Can You Beat Treewidth? *Theory Comput.*, 6(1):85–112, 2010. doi:10.4086/toc.2010.v006a005.

34 Dániel Marx. Tractable Hypergraph Properties for Constraint Satisfaction and Conjunctive Queries. *J. ACM*, 60(6):42:1–42:51, 2013. doi:10.1145/2535926.

35 Kitty Meeks. The challenges of unbounded treewidth in parameterised subgraph counting problems. *Discrete Appl. Math.*, 198:170–194, 2016. doi:10.1016/j.dam.2015.06.019.

36 R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network Motifs: Simple Building Blocks of Complex Networks. *Science*, 298(5594):824–827, 2002. doi:10.1126/science.298.5594.824.

37 Ron Milo, Shalev Itzkovitz, Nadav Kashtan, Reuven Levitt, Shai Shen-Orr, Inbal Ayzenshtat, Michal Sheffer, and Uri Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663):1538–1542, 2004. doi:10.1126/science.1089167.

38 Jaroslav Nešetřil and Patrice Ossona de Mendez. On nowhere dense graphs. *Eur. J. Comb.*, 32(4):600–617, 2011. doi:10.1016/j.ejc.2011.01.006.

39 Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity – Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-27875-4.

40 Jaroslav Nešetřil and Svatopluk Poljak. On the complexity of the subgraph problem. *Comment. Math. Univ. Carolin.*, 26(2):415–419, 1985. URL: https://eudml.org/doc/17394.

41 Norbert Peyerimhoff, Marc Roth, Johannes Schmitt, Jakob Stix, Alina Vdovina, and Philip Wellnitz. Parameterized Counting and Cayley Graph Expanders. *SIAM J. Discrete Math.*, to appear.

42 Marc Roth, Johannes Schmitt, and Philip Wellnitz. Counting Small Induced Subgraphs Satisfying Monotone Properties. *SIAM J. Comput.*, 0(0):FOCS20–139–FOCS20–174, 2022. doi:10.1137/20M1365624.

43 Marc Roth and Philip Wellnitz. Counting and finding homomorphisms is universal for parameterized complexity theory. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2161–2180. SIAM, 2020. doi:10.1137/1.9781611975994.133.

44 Benjamin Schiller, Sven Jager, Kay Hamacher, and Thorsten Strufe. StreaM – A Stream-Based Algorithm for Counting Motifs in Dynamic Graphs. In *Proceedings of the 2nd International Conference on Algorithms for Computational Biology (AlCoB)*, pages 53–67. Springer International Publishing, 2015. doi:10.1007/978-3-319-21233-3_5.

45 Carsten Thomassen. On the presence of disjoint subgraphs of a specified type. *J. Graph Theory*, 12(1):101–111, 1988. doi:10.1002/jgt.3190120111.

46 Ngoc Hieu Tran, Kwok Pui Choi, and Louxin Zhang. Counting motifs in the human interactome. *Nature communications*, 4(1):1–8, 2013. doi:10.1038/ncomms3241.

47 Charalampos E. Tsourakakis, Jakub Pachocki, and Michael Mitzenmacher. Scalable motif-aware graph clustering. In *Proceedings of the 26th International Conference on World Wide Web (WWW)*, pages 1451–1460, 2017. doi:10.1145/3038912.3052653.

48 Leslie G. Valiant. The Complexity of Enumeration and Reliability Problems. *SIAM J. Comput.*, 8(3):410–421, 1979. doi:10.1137/0208032.