COUNTING SUBGRAPHS IN SOMEWHERE DENSE GRAPHS*

MARCO BRESSAN[†], LESLIE ANN GOLDBERG[‡], KITTY MEEKS[§], AND MARC ROTH [¶]

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.

Key words. counting problems, somewhere dense graphs, parameterised complexity theory

MSC codes. 68Q17, 68Q25, 68R10

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 [50, 3, 10, 40] and to deep structural results in parameterised complexity theory [28, 18]. Since it was discovered that subgraph counts reveal global properties of complex networks [46, 47], subgraph counting has also found several applications in fields such as biology [2, 57] genetics [59], phylogeny [41], and data mining [60]. 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

*Preprint available at https://arxiv.org/abs/2209.03402. A preliminary version of this paper appeared in the Proceedings of the 14th Conference on Innovations in Theoretical Computer Science (ITCS 2023) [8]. This research was funded in whole, or in part, by EPSRC grant EP/V032305/1. All data is provided in full in the results section of this paper.

[†]Department of Computer Science, University of Milan, Italy

[‡]Department of Computer Science, University of Oxford, United Kingdom

[§]School of Computing Science, University of Glasgow, United Kingdom

[¶]School of Electronic Engineering and Computer Science, Queen Mary University of London, United Kingdom

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 [28, 23, 14, 20, 39, 45, 18, 55, 30].

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 [22]: 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 (see Section 2 for a complete introduction). 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 [28]. 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. 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 [48]. 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:[1]

THEOREM 1.1 (Theorem 18.9 in [49]). *If $\mathcal{G}$ is nowhere dense then $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$, $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$, 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.*

Thus the case of nowhere dense $\mathcal{G}$ 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 $\mathcal{G}$ is somewhere dense?

[1] In the realm of decision problems, an even more general meta-theorem is known for first-order model-checking on nowhere dense graphs [36].

1.1. Our Results. We prove dichotomies for #Sub($\mathcal{H} \to \mathcal{G}$), #IndSub($\mathcal{H} \to \mathcal{G}$), and #Hom($\mathcal{H} \to \mathcal{G}$) into FPT and #W[1]-hard cases, assuming that $\mathcal{G}$ is somewhere dense. It is known [56] 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 [18]. 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 section presents our main conceptual contribution (Section 1.1.1), gives a detailed walk-through of our complexity dichotomies (Section 1.1.2, Section 1.1.3, Section 1.1.4), provides some context (Section 1.2), and overviews the techniques behind our proofs (Section 1.3). For full proofs of our claims see Section 2 onward.

Basic preliminaries.. We concisely state some necessary definitions and observations which are given in more detail in Section 2. 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) [38]; 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.[2] All of our #W[1]-hardness results are actually #W[1]-completeness results; this holds because #Sub($\mathcal{H} \to \mathcal{G}$), #IndSub($\mathcal{H} \to \mathcal{G}$), and #Hom($\mathcal{H} \to \mathcal{G}$) are always in #W[1] due to a characterisation of #W[1] via parameterised model-counting problems (see [29, Chapter 14]).

1.1.1. 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 [28], the first proof required nine years and relied on sophisticated algebraic techniques [15]. This partially changed in 2017 when Curticapean, Dell and Marx [18] showed how to express a subgraph count $\#\mathsf{Sub}(H \to G)$ as linear combination of homomorphism counts $\sum_F a_F \cdot \#\mathsf{Hom}(F \to G)$. They showed that computing this 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

[2]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?" [43, 44].

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.[3] 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 [18] 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 1.3 gives a concrete example using the problem of counting k-matchings). This is why the technique of [18] 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 1.3:

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.

1.1.2. The Complexity of $\#\mathbf{Sub}(\mathcal{H} \to \mathcal{G})$.

This section presents our results on the fixed-parameter tractability of $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$. We start by presenting a minimal[4] 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 [28], $\#\textsc{Match}(\mathcal{U})$ 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 [15, 20, 18]. 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 1.2. *Let $\mathcal{G}$ be a monotone class of graphs[5] and assume that ETH holds. Then $\#\textsc{Match}(\mathcal{G})$ is fixed-parameter tractable if and only if $\mathcal{G}$ is nowhere dense.*

[3] 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 [11].

[4] 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".

[5] 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.

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* #W[1]*-hard and cannot be solved in time* $f(k)\cdot|G|^{o(k/\log k)}$ *for any function* f*.*

Theorem 1.2 subsumes the existing intractability results for counting k-matchings in bipartite graphs [20], in F-colourable graphs [56], in bipartite graphs with one-sided degree bounds [19], and in degenerate graphs [9]. It also strengthens the latter result: while [9] establishes hardness of counting k-matchings in ℓ-degenerate graphs for $k+\ell$ as a parameter, Theorem 1.2 yields hardness for d-degenerate graphs for every fixed $d \geq 2$.[6] Additionally, we show that Theorem 1.2 cannot be strengthened to achieve polynomial-time tractability of $\#\textsc{Match}(\mathcal{G})$ for nowhere dense and monotone $\mathcal{G}$, unless #P = P.

As a consequence of Theorem 1.2 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 1.3. *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.*

	$\mathcal{G}$ 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

Table 1

The complexity of $\#\textsc{Sub}(\mathcal{H}\to\mathcal{G})$ *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* $f(|H|)\cdot|G|^{o(|V(H)|)}$*.*

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. We point out that the contributions in this work are the hardness results in the third and fourth column, that is, for the cases in which $\mathcal{G}$ is somewhere dense, but not the class of all graphs. (For monotone $\mathcal{G}$, $\omega(\mathcal{G})=\infty$ implies that $\mathcal{G}$ is the class of all graphs.)

From the classification of Theorem 1.3 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 [20].

Theorem 1.4. *Let $\mathcal{H}$ and $\mathcal{G}$ be monotone classes of graphs and assume that ETH holds. Then* $\#\textsc{Sub}(\mathcal{H}\to\mathcal{G})$ *is fixed-parameter tractable if* $\mathrm{m}(\mathcal{H})<\infty$ *or $\mathcal{G}$ is nowhere dense; otherwise* $\#\textsc{Sub}(\mathcal{H}\to\mathcal{G})$ *is* #W[1]*-complete and cannot be solved in time*

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

$f(|H|) \cdot |G|^{o(|V(H)|/\log(|V(H)|))}$ *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 1.1, based on Gaifman's locality theorem for first-order formulas and the local sparsity of nowhere dense graphs (see [49]).
2. If $\mathrm{m}(\mathcal{H}) < \infty$ (first row of Table 1), then one can use the polynomial-time algorithm of Curticapean and Marx [20], 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.

1.1.3. The Complexity of #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 $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$. In this section we show that k-independent sets play a similar role for $\#\textsc{IndSub}(\mathcal{H} \to \mathcal{G})$. Let $\#\textsc{IndSet}(\mathcal{G}) = \#\textsc{IndSub}(\mathcal{I} \to \mathcal{G})$ where $\mathcal{I}$ is the set of all all independent sets (or 0-regular graphs). We prove:

Theorem 1.5. *Let* $\mathcal{G}$ *be a monotone class of graphs and assume that ETH holds. Then* $\#\textsc{IndSet}(\mathcal{G})$ *is fixed-parameter tractable if and only if* $\mathcal{G}$ *is nowhere dense. More precisely, if* $\mathcal{G}$ *is nowhere dense then* $\#\textsc{IndSet}(\mathcal{G})$ *can be solved in time* $f(k) \cdot |V(G)|^{1+o(1)}$ *for some computable function* f*; otherwise* $\#\textsc{IndSet}(\mathcal{G})$ *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 [17]. It also strengthens the result of [9], which shows $\#\textsc{IndSet}(\mathcal{G})$ is hard when parameterized by $k+d$ where d is the degeneracy of G. More precisely, [9] does not imply that $\#\textsc{IndSet}(\mathcal{G})$ is hard when $\mathcal{G}$ is the class of d-degenerate graphs, for any $d \geq 2$. In contrast to this, Theorem 1.5 proves such hardness for *every* $d \geq 2$. Finally, we point out that the FPT case of Theorem 1.5 is not in P unless P = #P.

As consequence of Theorem 1.5, when $\mathcal{H}$ is hereditary (and thus in particular monotone) we obtain:

Theorem 1.6. *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* $\#\textsc{IndSub}(\mathcal{H} \to \mathcal{G})$ *is exhaustively classified by Table 2.*

1.1.4. The Complexity of #Hom($\mathcal{H} \to \mathcal{G}$). Finally, we study the parameterized complexity of $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$. 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 [22, Chapter 7]. We prove:

	$\mathcal{G}$ 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

TABLE 2

The complexity of $\#\textsc{IndSub}(\mathcal{H} \to \mathcal{G})$ *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* $f(|H|) \cdot |G|^{o(|V(H)|)}$*.*

THEOREM 1.7. *Let* $\mathcal{H}$ *and* $\mathcal{G}$ *be monotone classes of graphs.*

1. *If* $\mathcal{G}$ *is nowhere dense then* $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$ *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* $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$ *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* $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$ *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 1.7 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 $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$ is easy since $|V(H)| \le 2$ or $\#\mathsf{Hom}(H \to G) = 0$. More generally, the complexity of $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$ 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 [9, 4]. There is some evidence that the finiteness of induced grid minors is the right criterion for tractability [9].

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

1.2. Related Work. The general idea of using interpolation as a reduction technique for counting problems dates back to the foundational work of Valiant [61]. 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. [42, Chapters 5 and 6]). It was then discovered by Curticapean et al. in [18] 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 the discussion in Section 1.1.1; 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 [5, 6, 4]. For example, [5] 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 sophisticated 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.

1.3. Overview of Our Techniques. The present section expands upon Section 1.1.1 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 1.3.1) and for $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$ (Section 1.3.2). The main contribution of our work is these hardness proofs. The upper bounds hold from (simple adaptations of) previous work.

1.3.1. 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 [27]. 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 1.2. From [18], 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 $G \in \mathcal{U}$,

$$\#\mathsf{Sub}(M_k \to G) = \sum_H a_k(H) \cdot \#\mathsf{Hom}(H \to G) \tag{1.1}$$

where the sum is over all isomorphism classes of all graphs. By a classic result of Dalmau and Jonsson [23], computing $\#\mathsf{Hom}(H \to G)$ is not fixed-parameter tractable for H of unbounded treewidth, unless ETH fails. Hence, if we could use (1.1) 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 [18] indeed prove. The idea is to apply (1.1) 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 (1.1).

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 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',

$$\#\mathsf{Hom}(H \to G \times G') = \#\mathsf{Hom}(H \to G) \cdot \#\mathsf{Hom}(H \to G'). \tag{1.2}$$

By combining (1.1) and (1.2), for each $i = 1, \ldots, \ell$ we obtain

$$\#\mathsf{Sub}(M_k \to G \times G_i) = \sum_{H} a_k(H) \cdot \#\mathsf{Hom}(H \to G_i) \cdot \#\mathsf{Hom}(H \to G) = \sum_{\substack{H \\ a_k(H)\neq 0}} b_H^i \cdot X_H\,, \tag{1.3}$$

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 compute a system of ℓ linear equations with the X_H as unknowns. By applying classical results due to Lovász (see e.g. [42, Chapter 5]), Curticapean et al. [18] 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 [18] 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 [23], 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 (1.2), 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. [20, 51, 26, 30]). 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 Section 2. 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 $\#\text{Hom}(\mathcal{E} \to \mathcal{U})$, which is known to be hard, to $\#\text{Hom}(\hat{\mathcal{F}} \to \mathcal{U}^r)$, where $\mathcal{U}^r$ is the set of all r-subdivisions of graphs. As noted above $\mathcal{U}^r \subseteq \mathcal{G}$, hence $\#\text{Hom}(\hat{\mathcal{F}} \to \mathcal{G})$ is hard. We will show in the coloured version that for each graph $F_k \in \hat{\mathcal{F}}$ with k edges, $a_k(F_k) \neq 0$ (see the proof of Lemma 4.6). 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) (1.3) 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 [51] and we describe them in Section 2.1.

Together, our resolutions of (I.1) and (I.2) yield a colourful version of the framework of [18] that applies to *any monotone somewhere dense class of host graphs*. As a consequence we obtain that $\#\text{Hom}(\mathcal{E} \to \mathcal{U})$, the problem of counting homomorphisms from expanders in $\mathcal{E}$ to arbitrary hosts graphs, reduces in FPT time to $\#\text{Sub}(\mathcal{M} \to \mathcal{G})$ whenever $\mathcal{G}$ is monotone and somewhere dense. Since $\#\text{Hom}(\mathcal{E} \to \mathcal{U})$ is intractable, this proves the hardness of $\#\text{Sub}(\mathcal{M} \to \mathcal{G})$ for all monotone and somewhere dense $\mathcal{G}$, as stated in Theorem 1.2. From this result we will then be able to prove our general classification for $\#\text{Sub}(\mathcal{H} \to \mathcal{G})$ (Theorem 1.3) by combining existing results and Ramsey-type arguments on $\mathcal{H}$ and $\mathcal{G}$.

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

1.3.2. Classifying Homomorphism Counting via Wall Minors. The proof of our dichotomy for $\#\text{Hom}(\mathcal{H} \to \mathcal{G})$ for monotone $\mathcal{H}$ and $\mathcal{G}$ (Theorem 1.7) 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) $\#\text{Hom}(\mathcal{E} \to \mathcal{U})$ to (the coloured version of) $\#\text{Hom}(\hat{\mathcal{F}} \to \mathcal{U}^r)$, 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.

To circumvent this problem, we use a result of Thomassen [58] to prove that, for every positive integer r, every monotone class of graphs $\mathcal{H}$ with unbounded treewidth, and every *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 $\#\text{Hom}(\mathcal{W} \to \mathcal{U})$ is intractable by the classification of Dalmau and Jonsson [23]. Refining our constructions based on subdivided graphs, we are then able to show that $\#\text{Hom}(\mathcal{W} \to \mathcal{U})$ reduces to $\#\text{Hom}(\mathcal{H} \to \mathcal{G})$ whenever $\mathcal{H}$ is monotone and of unbounded treewidth, and $\mathcal{G}$ is monotone and somewhere dense. Theorem 1.7 will then follow as a direct consequence.

2. Preliminaries. We denote the set of non-negative integers by $\mathbb{N}_0$, and the set of positive integers by $\mathbb{N}$. Graphs in this work are undirected and without self-loops unless stated otherwise. A *subdivision* of a graph G is obtained by subdividing

FIG. 1. *A fractured graph $Q\sharp\sigma$ from [51]. Left: a vertex $v \in V(Q)$ with incident edges $E_Q(v) = \{$●, ●, ●, ●, ●, ●$\}$. Right: the splitting of v in $Q\sharp\sigma$ for a fracture σ where the partition σ_v of $E_Q(v)$ consists of the two blocks $B_1 = \{$●, ●, ●$\}$, and $B_2 = \{$●, ●, ●$\}$.*

each edge of G arbitrarily often. Given a graph G and $r \in \mathbb{N}_0$, we write G^r for the *r-subdivision* of G, i.e., the graph obtained from G by subdividing each edge r times (so that it becomes a path of $r+1$ edges). Note that $G^0 = G$. (The graph G^{r-1} is also called the "r-stretch of G" in the literature). Given a graph G and a vertex $v \in V(G)$, we write $E_G(v) := \{e \in E(G) \mid v \in e\}$ for the set of edges incident to v. Furthermore, given $A \subseteq E(G)$, we write $G[A]$ for the graph $(V(G), A)$. Given a subset of vertices $S \subseteq V(G)$, we write $G[S]$ for the subgraph of G induced by the vertices in S, that is, $G[S] := (S, \{e \in E(G) \mid e \subseteq S\})$. An "induced subgraph" of G is a subgraph induced by some $S \subseteq V(G)$.

A *homomorphism* from a graph H to a graph G is a mapping $\varphi : V(H) \to V(G)$ which is edge-preserving, that is, $\{u, v\} \in E(H)$ implies $\{\varphi(u), \varphi(v)\} \in E(G)$. We write:

- $\mathsf{Hom}(H \to G)$ for the set of all homomorphisms from H to G,
- $\mathsf{SurHom}(H \to G)$ for the set of all surjective homomorphisms from H to G,
- $\mathsf{Sub}(H \to G)$ for the set of all subgraphs of G isomorphic to H, and
- $\mathsf{IndSub}(H \to G)$ for the set of all induced subgraphs of G isomorphic to H.

2.1. Coloured Graphs and Fractures. Let H be a graph. Following standard terminology, we refer to an element of $\mathsf{Hom}(G \to H)$ as an *H-colouring* of the graph G. An *H-coloured graph* is a pair (G, c) where G is a graph and c an H-colouring of G. We say that (G, c) is a surjectively H-coloured graph if $c \in \mathsf{SurHom}(G \to H)$.

Given two H-coloured graphs (F, c_F) and (G, c_G), a homomorphism from (F, c_F) to (G, c_G) is a mapping $\varphi \in \mathsf{Hom}(F \to G)$ such that $c_G(\varphi(v)) = c_F(v)$ for each $v \in V(F)$.[7] We write $\mathsf{Hom}((F, c_F) \to (G, c_G))$ for the set of all homomorphisms from (F, c_F) to (G, c_G).

Following the terminology of [51], we define a fracture of a graph H as a $|V(H)|$-tuple $\rho = (\rho_v)_{v \in V(H)}$ where ρ_v is a partition of the set $E_H(v)$ of edges of H incident to v. Now, given a fracture ρ of H, we obtain the *fractured graph* $H\sharp\rho$ from H by splitting each vertex v according to the partition ρ_v. Formally, the graph $H\sharp\rho$ contains a vertex v^B for each vertex $v \in V(H)$ and block $B \in \rho_v$, and we make v^B and $u^{B'}$ adjacent if and only if $\{v, u\} \in E(H)$ and $\{u, v\} \in B \cap B'$. An illustration is provided in Figure 1.

The following H-colouring of a fractured graph is used implicitly in [51].

DEFINITION 2.1. *Let H be a graph and ρ a fracture of H. We denote by $c_\rho : V(H\sharp\rho) \to V(H)$ the function that maps v^B to v for each $v \in V(H)$ and $B \in \rho_v$.*

[7] We remark that in previous work [51], homomorphisms between H-coloured graphs are called "colour-preserving" or, if $F = H$, "colour-prescribed". Since we will work almost exclusively in the coloured setting in this work, we will just speak of homomorphisms and always provide the H-colourings explicitly in our notation.

Observation 2.2. *For each H and ρ, c_ρ is an H-colouring of $H \sharp \rho$.*

2.2. Graph Classes, Invariants and Minors. We use symbols $\mathcal{F}, \mathcal{G}, \mathcal{H}$ to denote classes of graphs, and we denote by $\mathcal{U}$ be the class of *all* graphs. A graph invariant is a function $g : \mathcal{U} \to \mathbb{N}_0$ such that $g(G) = g(H)$ whenever G and H are isomorphic. An invariant g is *bounded* on a graph family $\mathcal{H}$ if there exists $B \in \mathbb{N}_0$ such that $g(H) \leq B$ for all $H \in \mathcal{H}$, in which case we write $g(\mathcal{H}) < \infty$; otherwise we say g is unbounded on $\mathcal{H}$ and write $g(\mathcal{H}) = \infty$. Our statements involve the following invariants.

Definition 2.3 (Graph Invariants). *For any graph G define:*

- *the independence number $\alpha(G)$, i.e., the size of the largest independent set of G*
- *the clique number $\omega(G)$, i.e., the size of the largest complete subgraph of G*
- *the biclique number $\beta(G)$, i.e., the largest k such that G contains a k-by-k biclique as a subgraph, and its induced version, the induced biclique number $\beta_{\mathsf{ind}}(G)$*
- *the matching number $\mathrm{m}(G)$, i.e., the size of a maximum matching of G, and its induced version, the induced matching number $\mathrm{m}_{\mathsf{ind}}(G)$*

We denote by $\mathsf{tw}(G)$ the *treewidth* of a graph G. We omit the definition of treewidth as we rely on it in a black-box manner; the interested reader can see e.g. Chapter 7 of [22]. For any $k \in \mathbb{N}$ the k-by-k grid graph $\boxplus_k$, depicted in Figure 2, is defined by $V(\boxplus_k) = [k]^2$ and $E(\boxplus_k) = \{\{(i,j),(i',j')\} : i,j,i',j' \in [k], |i-i'| + |j-j'| = 1\}$. It is well known that $\mathsf{tw}(\boxplus_k) = k$, see [22, Chapter 7.7.1].

We make use of the following two consequences of Ramsey's Theorem for an arbitrary class of graphs $\mathcal{H}$. The first one is immediate, and the second one was established by Curticapean and Marx in [20].

Theorem 2.4. *If $|\mathcal{H}| = \infty$ then $\max(\alpha(\mathcal{H}), \omega(\mathcal{H})) = \infty$.*

Theorem 2.5. *If $\mathrm{m}(\mathcal{H}) = \infty$ then $\max(\omega(\mathcal{H}), \beta_{\mathsf{ind}}(\mathcal{H}), \mathrm{m}_{\mathsf{ind}}(\mathcal{H})) = \infty$.*

A class of graphs is *hereditary* if it is closed under vertex deletion, and is *monotone* if it is hereditary and closed under edge deletion. In other words, hereditary classes are closed under taking induced subgraphs, and monotone classes are closed under taking subgraphs.

To present the different notions of graph minors used in this paper in a unified way, we start by introducing *contraction models.*

Definition 2.6 (Contraction model). *A* contraction model *of a graph H in a graph G is a partition $\{V_1, \dots, V_k\}$ of $V(G)$ such that $G[V_i]$ is connected for each $i \in [k]$ and that H is isomorphic to the graph obtained from G by contracting each $G[V_i]$ into a single vertex (and deleting multiple edges and self-loops).*

Recall that a graph F is a *minor* of a graph G if F can be obtained from G by deletion of edges and vertices, and by contraction of edges; equivalently, F is a minor of G if F is a subgraph of a graph that has a contraction model in G. In this work, we will also require the subsequent stricter notion of minors.

Definition 2.7 (Shallow minor [48]). *A graph F is a* shallow minor *at depth d of a graph G if F is a subgraph of graph H that has a contraction model $\{V_1, \dots, V_k\}$ in G satisfying the following additional constraint: for each $i \in [k]$ there is a vertex $x_i \in V_i$ such that each vertex in V_i has distance at most d from x_i. Given a class of graphs $\mathcal{G}$, we write $\mathcal{G} \triangledown d$ for the set of all shallow minors at depth d of graphs in $\mathcal{G}$.*

Observe that the shallow minors at depth 0 of G are exactly the subgraphs of G, and the shallow minor of depth $|V(G)|$ are exactly the minors of G. For this reason, the notion of a shallow minor can be considered an interpolation between subgraphs and minors. Furthermore, having introduced this notion, we are now able to define somewhere dense and nowhere dense graph classes.

DEFINITION 2.8 (Somewhere dense graph classes [48]). *A class of graphs $\mathcal{G}$ is* somewhere dense *if $\omega(\mathcal{G}\nabla d) = \infty$ for some $d \in \mathbb{N}_0$, and is* nowhere dense *if instead $\omega(\mathcal{G}\nabla d) < \infty$ for all $d \in \mathbb{N}_0$.*

We use the following characterisation of monotone somewhere dense graph classes.[8]

LEMMA 2.9 (Remark 2 in [1]). *Let $\mathcal{G}$ be a monotone class of graphs. Then $\mathcal{G}$ is somewhere dense if and only if there exists $r \in \mathbb{N}_0$ such that $G^r \in \mathcal{G}$ for all $G \in \mathcal{U}$.*

2.3. Parameterised and Fine-Grained Complexity. A *parameterized counting problem* is a pair (P, κ) where $P : \{0,1\}^* \to \mathbb{N}$ and $\kappa : \{0,1\}^* \to \mathbb{N}$ is computable. For an instance x of P we call $\kappa(x)$ the *parameter* of x. An algorithm $\mathbb{A}$ is *fixed-parameter tractable* (FPT) w.r.t. a parameterization κ if there is a computable function f such that $\mathbb{A}$ runs in time $f(\kappa(x)) \cdot |x|^{O(1)}$ on every input x. A parameterized counting problem (P, κ) is *fixed-parameter tractable* (FPT) if there is an FPT algorithm (w.r.t. κ) that computes P.

A *parameterized Turing reduction* from (P, κ) to (P', κ') is an algorithm $\mathbb{A}$ equipped with oracle access to P' satisfying the following constraints:

(A1) $\mathbb{A}$ computes P

(A2) $\mathbb{A}$ is FPT w.r.t. κ

(A3) there is a computable function g such that, on input x, each oracle query x' satisfies that $\kappa'(x') \leq g(\kappa(x))$.

We write $(P, \kappa) \leq^{\mathsf{FPT}} (P', \kappa')$ if a parameterized Turing reduction from (P, κ) to (P', κ') exists.

The parameterized counting problem #CLIQUE asks, on input a graph G and $k \in \mathbb{N}$, to compute the number of k-cliques in G; the parameter is k. As shown by Flum and Grohe [28], #CLIQUE is the canonical complete problem for the parameterized complexity class #W[1]. In particular, a parameterized counting problem (P, κ) is called #W[1]-*hard* if #CLIQUE $\leq^{\mathsf{FPT}} (P, \kappa)$. We omit the technical definition of #W[1] via weft-1 circuits (see Chapter 14 of [29]), but we recall that #W[1]-hard problems are not FPT unless standard hardness assumptions fail (see below). We define the problems studied in this work. As usual $\mathcal{H}$ and $\mathcal{G}$ denote classes of graphs.

DEFINITION 2.10. #HOM$(\mathcal{H} \to \mathcal{G})$, #SUB$(\mathcal{H} \to \mathcal{G})$, #INDSUB$(\mathcal{H} \to \mathcal{G})$ *ask, given $H \in \mathcal{H}$ and $G \in \mathcal{G}$, to compute respectively* $\#\mathsf{Hom}(H \to G)$, $\#\mathsf{Sub}(H \to G)$, *and* $\#\mathsf{IndSub}(H \to G)$. *The parameter is $|H|$.*

For example, #SUB$(\mathcal{H} \to \mathcal{G})$ = #CLIQUE when $\mathcal{H}$ is the class of all complete graphs and $\mathcal{G}$ the class of all graphs. The following result follows immediately from an algorithm for counting answers to Boolean queries in nowhere dense graphs due to Nešetřil and Ossona de Mendez [49].

THEOREM 2.11 (Theorem 18.9 in [49]). *If $\mathcal{G}$ is nowhere dense then* #HOM$(\mathcal{H} \to \mathcal{G})$, #SUB$(\mathcal{H} \to \mathcal{G})$, *and* #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.*

[8]It is non-trivial to pinpoint the first statement of Lemma 2.9 in the literature: Dvorák et al. [27] attribute it to Nešetřil and de Mendez [48], who provide an implicit proof. The first explicit statement is, to the best of our knowledge, due to Adler and Adler [1].

In an intermediate step towards our classifications, we will rely on a coloured version of homomorphism counting.

Definition 2.12. $\#\text{CP-HOM}(\mathcal{H} \to \mathcal{G})$ *asks, given* $H \in \mathcal{H}$ *and a surjectively*[9] H*-coloured graph* (G, c) *with* $G \in \mathcal{G}$*, to compute* $\#\mathsf{Hom}((H, \mathsf{id}_H) \to (G, c))$*, where* id_H *denotes the identity on* $V(H)$*. The parameter is* $|H|$*.*

It is well known that $\#\text{CP-HOM}(\mathcal{H} \to \mathcal{U})$ reduces to the uncoloured version via inclusion-exclusion. The same holds for $\#\text{CP-HOM}(\mathcal{H} \to \mathcal{G})$, too, if $\mathcal{G}$ is monotone. Formally:

Lemma 2.13 (see e.g. Lemma 2.49 in [53]). *If* $\mathcal{G}$ *is monotone then* $\#\text{CP-HOM}(\mathcal{H} \to \mathcal{G}) \leq^{\mathsf{FPT}} \#\text{HOM}(\mathcal{H} \to \mathcal{G})$*. Moreover, on input* $H \in \mathcal{H}$ *and* (G, c) *with* $G \in \mathcal{G}$*, every oracle query* (H', G') *in the reduction satisfies* $H' = H$ *and* $G' \subseteq G$*.*

An implicit consequence of the parameterized complexity classification for counting homomorphisms due to Dalmau and Jonsson [23] establishes the following hardness result for $\#\text{CP-HOM}$; an explicit argument can be found e.g. in Chapter 2 in [53].

Theorem 2.14 ([23]). *If* $\mathcal{H}$ *is recursively enumerable and* $\mathsf{tw}(\mathcal{H}) = \infty$ *then* $\#\text{CP-HOM}(\mathcal{H} \to \mathcal{U})$ *is* $\#\mathrm{W}[1]$*-hard.*

Finally, all running-time lower bounds in this paper are conditional on ETH:

Definition 2.15 ([38]). *The* Exponential Time Hypothesis (ETH) *asserts that* 3-SAT *cannot be solved in time* $\mathsf{exp}(o(n))$ *where* n *is the number of variables of the input formula.*

Chen et al. [12, 13] showed that there is no function f such that $\#\text{CLIQUE}$ can be solved in time $f(k) \cdot |G|^{o(k)}$ unless ETH fails. This in particular implies that $\#\mathrm{W}[1]$-hard problems are not FPT unless ETH fails. Marx [43] strengthened Theorem 2.14 into:[10]

Theorem 2.16 ([43]). *If* $\mathcal{H}$ *is recursively enumerable and* $\mathsf{tw}(\mathcal{H}) = \infty$ *then* $\#\text{CP-HOM}(\mathcal{H} \to \mathcal{U})$ *cannot be solved in time* $f(|H|) \cdot |G|^{o\left(\frac{\mathsf{tw}(H)}{\log \mathsf{tw}(H)}\right)}$ *for any function* f*, unless ETH fails.*

The question of whether the $(\log \mathsf{tw}(H))^{-1}$ factor in the above lower bound can be omitted can be considered the counting version of the open problem "Can you beat treewidth?" [43, 44].

3. Counting Homomorphisms.

This section is devoted to the proof of our dichotomy theorem for $\#\text{HOM}(\mathcal{H} \to \mathcal{G})$, Theorem 1.7. We start by showing a reduction from $\#\text{CP-HOM}(\mathcal{H} \to \mathcal{U})$ to counting colour-prescribed homomorphisms between subdivided graphs. While the proof is straightforward, the reduction will turn out useful for the more involved cases of $\#\text{SUB}(\mathcal{H} \to \mathcal{G})$ and $\#\text{INDSUB}(\mathcal{H} \to \mathcal{G})$. Theorem 1.7 will be an immediate consequence of Theorem 2.11 and Theorem 3.6 below.

To begin with, let $c \in \mathsf{SurHom}(G \to H)$ and let $r \in \mathbb{N}_0$. Define the following canonical homomorphism c^r from G^r to H^r. For each $u \in V(G)$, set $c^r(u) := c(u)$. For any edge $e = \{u_1, u_2\} \in E(G)$, let $u_1, w_1, \ldots, w_r, u_2$ be the corresponding path in G^r. Let $e' = \{v_1, v_2\} = \{c(u_1), c(u_2)\}$ — note that $e' \in E(H)$ as $c \in \mathsf{Hom}(G \to H)$

[9] In previous works (e.g. in [51]), the definition of $\#\text{CP-HOM}(\mathcal{H} \to \mathcal{G})$ did not require the H-colouring to be surjective. However, one can always assume surjectivity, since $\#\mathsf{Hom}((H, \mathsf{id}_H) \to (G, c)) = 0$ if c is not surjective. We decided to make the surjectivity condition explicit in this work.

[10] More precisely, Marx established the bound for the so-called partitioned subgraph problem. However, as shown in [55], the lower bound immediately translates to $\#\text{CP-HOM}(\mathcal{H} \to \mathcal{U})$.

— and let $v_1, x_1, \dots, x_r, v_2$ be the corresponding path in H^r. Then, set $c^r(w_i) := x_i$ for each $i \in \{1, \dots, r\}$. It is easy to see that c^r is a surjective H^r-colouring of G^r. Furthermore:

LEMMA 3.1. *For every surjectively H-coloured graph (G, c) and every $r \in \mathbb{N}_0$,*

$$\#\mathsf{Hom}((H, \mathsf{id}_H) \to (G, c)) = \#\mathsf{Hom}((H^r, \mathsf{id}_{H^r}) \to (G^r, c^r)) \tag{3.1}$$

where id_H and id_{H^r} are the identities on respectively $V(H)$ and $V(H^r)$.

Proof. We define a bijection $b : \mathsf{Hom}((H, \mathsf{id}_H) \to (G, c)) \to \mathsf{Hom}((H^r, \mathsf{id}_{H^r}) \to (G^r, c^r))$. Let $\varphi \in \mathsf{Hom}((H, \mathsf{id}_H) \to (G, c))$. For every $v \in V(H)$ let $b(\varphi)(v) = \varphi(v)$. For every $\{v_1, v_2\} \in E(H)$ and every $i \in [r]$, if $u_1 = \varphi(v_1)$ and $u_2 = \varphi(v_2)$, and if x_i and w_i are the i-th vertices on the paths respectively between v_1 and v_2 in H^r and between u_1 and u_2 in G^r, then let $b(\varphi)(x_i) = w_i$. It is easy to see that $b(\varphi) \in \mathsf{Hom}((H^r, \mathsf{id}_{H^r}) \to (G^r, c^r))$ and that b is injective. To see that b is surjective as well, note that for every $\varphi^r \in \mathsf{Hom}((H^r, \mathsf{id}_{H^r}) \to (G^r, c^r))$ its restriction $\varphi^r|_{V(H)}$ to $V(H)$ satisfies $\varphi^r|_{V(H)} \in \mathsf{Hom}((H, \mathsf{id}_H) \to (G, c))$ and $b(\varphi^r|_{V(H)}) = \varphi^r$. ∎

3.1. Warm-up: Minor-closed Pattern Classes. Using the characterisation of somewhere dense graph classes in Lemma 2.9, and known lower bounds for counting homomorphisms from grid graphs, we obtain as an easy consequence the following complexity dichotomy:

THEOREM 3.2. *Let $\mathcal{H}$ be a minor-closed class of graphs and let $\mathcal{G}$ be a monotone and somewhere dense class of graphs.*

1. *If $\mathsf{tw}(\mathcal{H}) < \infty$ then $\#\text{HOM}(\mathcal{H} \to \mathcal{G}) \in \mathsf{P}$. Moreover, if a tree decomposition of H of width t is given, then $\#\text{HOM}(\mathcal{H} \to \mathcal{G})$ can be solved in time $|H|^{O(1)} \cdot |V(G)|^{t+1}$.*
2. *If $\mathsf{tw}(\mathcal{H}) = \infty$, then $\#\text{HOM}(\mathcal{H} \to \mathcal{G})$ is $\#\mathrm{W}[1]$-hard and, assuming ETH, cannot be solved in time $f(|H|) \cdot |G|^{o(\mathsf{tw}(H))}$ for any function f.*

Proof. The tractability result is well known [24, 23], so we only need to prove the hardness part. Recall that $\boxplus_k$ denotes the k-by-k grid; see Figure 2 for a depiction of $\boxplus_4$. Let $\boxplus := \{\boxplus_k \mid k \in \mathbb{N}\}$. It is known that $\#\text{CP-HOM}(\boxplus \to \mathcal{U})$ is $\#\mathrm{W}[1]$-hard and, unless ETH fails, cannot be solved in time $f(k) \cdot |G|^{o(k)}$ for any function f (see [16, Lemma 1.13 and 5.7] or [53, Lemma 2.45]). As $\mathsf{tw}(\boxplus_k) = k$, the lower bound above can be written as $f(k) \cdot |G|^{o(\mathsf{tw}(\boxplus_k))}$.

Let $(\boxplus_k, (G, c))$ be the input to $\#\text{CP-HOM}(\boxplus \to \mathcal{U})$. Since $\mathcal{G}$ is somewhere dense and monotone, by Lemma 2.9 there is $r \in \mathbb{N}_0$ such that $\mathcal{G}$ contains the r-subdivision of every graph and thus, in particular, G^r. Moreover, since $\mathsf{tw}(\mathcal{H}) = \infty$ and $\mathcal{H}$ is minor-closed, by the Excluded-Grid Theorem [52] $\mathcal{H}$ contains every planar graph and thus in particular $\boxplus_k^r$. Clearly, $\boxplus_k^r$, G^r and c^r can be computed in polynomial time. Moreover, by Lemma 3.1,

$$\#\mathsf{Hom}((\boxplus_k, \mathsf{id}_{\boxplus_k}) \to (G, c)) = \#\mathsf{Hom}((\boxplus_k^r, \mathsf{id}_{\boxplus_k^r}) \to (G^r, c^r)).$$

Hence $\#\text{CP-HOM}(\boxplus \to \mathcal{U}) \leq^{\mathsf{FPT}} \#\text{CP-HOM}(\mathcal{H} \to \mathcal{G})$. Since $\#\text{CP-HOM}(\mathcal{H} \to \mathcal{G}) \leq^{\mathsf{FPT}} \#\text{HOM}(\mathcal{H} \to \mathcal{G})$ by Lemma 2.13, we conclude that $\#\text{HOM}(\mathcal{H} \to \mathcal{G})$ is $\#\mathrm{W}[1]$-hard. For the conditional lower bound, observe that both reductions used above preserve the treewidth of the pattern (the first because treewidth is invariant under edge subdivision,[11] the second by Lemma 2.13). ∎

[11] For example, this invariance is in Exercises 7.7 and 7.13 in [22].

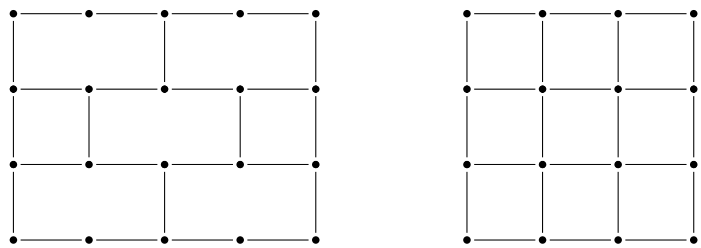

FIG. 2. *The wall $W_{4,5}$ (left) and the grid $\boxplus_4$ (right).*

3.2. Monotone Pattern Classes. The strengthening of Theorem 3.2 to monotone pattern classes can be done by reduction from counting homomorphisms from a class of well-known graphs called *walls*.

DEFINITION 3.3 (Walls). *Let $k, \ell \in \mathbb{N}$. The* wall *of height k and length ℓ, denoted by $W_{k,\ell}$, is the graph whose vertex set is $\{v_{i,j} : 1 \le i \le k, 1 \le j \le \ell\}$ and whose edge set contains:*

- *$\{v_{i,j}, v_{i,j+1}\}$ for all $1 \le i \le k$ and $1 \le j \le \ell - 1$,*
- *$\{v_{i,1}, v_{i+1,1}\}$ and $\{v_{i,\ell}, v_{i+1,\ell}\}$ for all $1 \le i \le k-1$*
- *$\{v_{i,j}, v_{i+1,j}\}$ for all $1 \le i \le k-1$ and $1 \le j \le \ell$ such that $i + j$ is even.*

Figure 2 depicts $W_{4,5}$ as an example. We let $\mathcal{W} := \{W_{k,k} \mid k \in \mathbb{N}\}$ be the class of all walls.

The following structural property of large walls is due to Thomassen.[12]

LEMMA 3.4 (Proposition 3.2 in [58]). *For every $k, r \in \mathbb{N}$, there exists $h(k,r) \in \mathbb{N}$ such that every subdivision of $W_{h(k,r),h(k,r)}$ contains as a subgraph a subdivision of $W_{k,k}$ in which each edge is subdivided a (positive) multiple of r times.*

The final ingredient of our proof for the classification of monotone pattern classes is given by Lemma 3.5, which is an immediate consequence of Lemma 2.9.

LEMMA 3.5. *Let $\mathcal{G}$ be a monotone and somewhere dense class of graphs. There exists $r \in \mathbb{N}_0$ such that the following holds. Let G be any graph and let G' be any graph obtained from G by subdividing each edge a (positive) multiple of r times. Then G' is contained in $\mathcal{G}$.*

Proof. We show that the claim holds for the $r \in \mathbb{N}_0$ given by Lemma 2.9. For this r, Lemma 2.9 guarantees that for every graph H, $H^r \in \mathcal{G}$. Now let G be any graph and label its edges $e_1, \dots, e_m$. Let G' be any graph obtained from G by subdividing each edge a (positive) multiple of r times. Then there exist $d_1, \dots, d_m \in \mathbb{N}$ such that, for each $i \in [m]$, the edge e_i is subdivided $d_i r$ times. Now let $\hat{G}$ be the graph obtained from G by subdividing, for each $i \in [m]$, the edge e_i just d_i times. It is immediate that $G' = \hat{G}^r$. Thus, by Lemma 2.9 and our choice of r, we have that $G' \in \mathcal{G}$. ∎

We are now ready to establish the main result of this section.

THEOREM 3.6. *Let $\mathcal{H}$ be a monotone class of graphs and let $\mathcal{G}$ be a monotone and somewhere dense class of graphs.*

1. *If $\mathsf{tw}(\mathcal{H}) < \infty$ then $\#\text{HOM}(\mathcal{H} \to \mathcal{G}) \in \mathsf{P}$. Moreover, if a tree decomposition of H of width t is given, then $\#\text{HOM}(\mathcal{H} \to \mathcal{G})$ can be solved in time $|H|^{O(1)} \cdot |V(G)|^{t+1}$.*

[12] Note that walls are called grids in [58].

2. *If* $\mathsf{tw}(\mathcal{H}) = \infty$*, then* $\#\text{Hom}(\mathcal{H} \to \mathcal{G})$ *is* #W[1]*-hard and, assuming ETH, cannot be solved in time* $f(|H|) \cdot |G|^{o(\mathsf{tw}(H))}$ *for any function* f*.*

Proof. The tractability result is well known [24, 23], so we only need to prove point 2. To this end, we will reduce from $\#\text{cp-Hom}(\mathcal{W} \to \mathcal{U})$. Walls clearly have grid minors of linear size, that is, there is a function $h \in \Theta(k)$ such that $W_{k,k}$ contains $\boxplus_{h(k)}$ as a minor. Furthermore, it is well-known that #cp-Hom is minor-monotone (see e.g. [16, Lemma 5.8] or [53, Lemma 2.47]), hence $\#\text{cp-Hom}(\boxplus \to \mathcal{U}) \leq^{\mathsf{FPT}} \#\text{cp-Hom}(\mathcal{W} \to \mathcal{U})$. Moreover the reduction is tight, in the sense that the lower bound for $\#\text{cp-Hom}(\boxplus \to \mathcal{U})$ shown in the proof of Theorem 3.2 transfers to $\#\text{cp-Hom}(\mathcal{W} \to \mathcal{U})$; hence $\#\text{cp-Hom}(\mathcal{W} \to \mathcal{U})$ is #W[1]-hard and, assuming ETH, it cannot be solved in time $f(k) \cdot |G|^{o(\mathsf{tw}(W_{k,k}))}$ for any function f.

Let us now construct the reduction $\#\text{cp-Hom}(\mathcal{W} \to \mathcal{U}) \leq^{\mathsf{FPT}} \#\text{cp-Hom}(\mathcal{H} \to \mathcal{G})$. Let $r \in \mathbb{N}_0$ as given by Lemma 3.5. We use the fact that $\mathsf{tw}(\mathcal{H}) = \infty$ implies that $\mathcal{H}$ contains as minors all planar graphs; that is, for every planar graph F there is a graph $H \in \mathcal{H}$ such that F is a minor of H [52]. In particular, $\mathcal{H}$ contains all walls $W_{k,k}$ as minors. A graph J is said to be a "topological minor" of a graph H if there is a subdivision of J that is isomorphic to a subgraph of H. Since walls have degree at most 3, the fact that $\mathcal{H}$ contains all walls as minors implies that it also contains all walls as topological minors (see e.g. [25, Proposition 1.7.3]).

Now let $W_{k,k}$ and (G, c) be an input instance of $\#\text{cp-Hom}(\mathcal{W} \to \mathcal{U})$. Let $e_1, \ldots, e_\ell$ be the edges of $W_{k,k}$ in arbitrary order. By Lemma 3.4, every subdivision of $W_{h(k,r),h(k,r)}$ contains as a subgraph a subdivision of $W_{k,k}$ in which each edge is subdivided a (positive) multiple of r times. Since $\mathcal{H}$ contains $W_{k,k}$ as a topological minor, there is a subdivision of $W_{k,k}$ that is isomorphic to a subgraph W' of a graph in $\mathcal{H}$. Since $\mathcal{H}$ is monotone, there are $W' \in \mathcal{H}$ and $d_1, \ldots, d_\ell \in \mathbb{N}_0$ such that W' is obtained from $W_{k,k}$ by subdividing e_i precisely $d_i r$ times for each $i \in [\ell]$.

We will now construct from (G, c) a graph G' and a surjective homomorphism c' from G' to W'. For each edge $e = \{u, v\}$ of G we proceed as follows. Since $c \in \mathsf{Hom}(G \to W_{k,k})$, then $\{c(u), c(v)\} = e_i$ for some $i \in [\ell]$. By the definition of W', e_i was replaced by a path $c(u), x_1, \ldots, x_{d_i r}, c(v)$. Hence, we replace the edge e in G by a path $u, w_1, w_2, \ldots, w_{d_i r}, v$, where the w_j are fresh vertices. Furthermore, we extend the colouring c to the colouring c' by setting $c'(w_j) := x_j$ for each $j \in [d_i r]$. Since c is surjective, so is c'. Also,

$$\#\mathsf{Hom}((W_{k,k}, \mathsf{id}_{W_{k,k}}) \to G, c) = \#\mathsf{Hom}((W', \mathsf{id}_{W'}) \to (G', c'))\,.$$

By querying the oracle for $\#\text{cp-Hom}(\mathcal{H} \to \mathcal{G})$ on the instance $((W', \mathsf{id}_{W'}), (G', c'))$ we can thus conclude our reduction. This immediately implies #W[1]-hardness of $\#\text{cp-Hom}(\mathcal{H} \to \mathcal{G})$. For the conditional lower bound, we observe that W' has the same treewidth as $W_{k,k}$ since it is a subdivision of $W_{k,k}$, and that the size of (G', c') is clearly bounded by $f(k) \cdot |G|^{O(1)}$ — note that the f depends on $\mathcal{H}$ which is, however, fixed. A reduction to the uncoloured version via Lemma 2.13 completes the proof. ◻

Theorem 1.7 follows immediately from Theorem 2.11 and Theorem 3.6. We conclude with a remark.

Remark 3.7. *A strengthening of Theorem 3.6 to hereditary pattern classes* $\mathcal{H}$ *is not possible. Suppose for instance that* $\mathcal{H}$ *contains all complete graphs and* $\mathcal{G}$ *is the class of all bipartite graphs. Although* $\mathcal{H}$ *is hereditary and of unbounded treewidth, and* $\mathcal{G}$ *is monotone and somewhere dense, it is easy to see that* $\#\text{Hom}(\mathcal{H} \to \mathcal{G})$ *is trivial, since we can always output zero if* $H \in \mathcal{H}$ *has at least* 3 *vertices. When it comes* *to a sufficient and necessary condition for tractability in case of hereditary classes of patterns, we conjecture that* induced grid minor size *might be the right candidate. However, even for very special cases, such as classes of degenerate host graphs (which are somewhere dense and monotone), it is still open whether induced grid minor size is the correct answer [9]. Thus, we leave the classification for hereditary classes of patterns as an open problem for further research.*

4. Counting Subgraphs.

This section is devoted to the proofs of Theorem 1.2, Theorem 1.4, and Theorem 1.3. We begin in Section 4.1 by analysing the problem of counting k-matchings in somewhere dense host graphs, and proving Theorem 1.2; this is the most technical part. We then move on to prove Theorem 1.4 and Theorem 1.3 in Section 4.2.

4.1. Counting Matchings: Proof of Theorem 1.2.

A k-matching in a graph G is a set $M \subseteq E(G)$ with $|M| = k$ and $e_1 \cap e_2 = \emptyset$ for all $e_1 \neq e_2$ in M. In other words, a k-matching in G is a set of k pairwise non-incident edges of G. Given a class of graphs $\mathcal{G}$, the problem $\#\textsc{Match}(\mathcal{G})$ asks, on input $k \in \mathbb{N}$ and a graph $G \in \mathcal{G}$, to compute the number of k-matchings in G; the parameter is k. We remark that $\#\textsc{Match}(\mathcal{G}) = \#\textsc{Sub}(\mathcal{M} \to \mathcal{G})$ where $\mathcal{M}$ is the set of all 1-regular graphs. The goal of this section is to prove that $\#\textsc{Match}(\mathcal{G})$ is hard whenever $\mathcal{G}$ is monotone and somewhere dense, i.e., the hardness part of Theorem 1.2.

Before moving on, let us pin down some definitions and basic facts. Our analysis relies on the following "coloured" version of the graph tensor product, as in [51]:

Definition 4.1. *Let H be a graph, and let (G_1, c_1) and (G_2, c_2) be H-coloured graphs. The* tensor product $(G_1, c_1) \times (G_2, c_2)$ *is the H-coloured graph $(\hat{G}, \hat{c})$ defined by:*

(T1) $V(\hat{G}) = \{(v_1, v_2) \in V(G_1) \times V(G_2) \mid c_1(v_1) = c_2(v_2)\}$.

(T2) $\{(u_1, u_2), (v_1, v_2)\} \in E(\hat{G})$ *if and only if* $\{u_1, v_1\} \in E(G_1)$ *and* $\{u_2, v_2\} \in E(G_2)$.

(T3) $\hat{c}(v_1, v_2) = c_1(v_1)$ *(equivalently by (T1),* $\hat{c}(v_1, v_2) = c_2(v_2)$*) for all* $(v_1, v_2) \in V(\hat{G})$.

The crucial property of the tensor product is given by:[13]

Lemma 4.2 ([51]). *If H is a graph and (F, c_F), (G_1, c_1), (G_2, c_2) are H-coloured graphs, then*

$$\#\mathsf{Hom}((F, c_F) \to (G_1, c_1) \times (G_2, c_2)) = \#\mathsf{Hom}((F, c_F) \to (G_1, c_1)) \cdot \#\mathsf{Hom}((F, c_F) \to (G_2, c_2)) \,.$$

The final ingredient we need is the non-singularity of a certain matrix whose entries count homomorphisms between fractured graphs. Formally, let H be a graph. The square matrix M_H has its rows and columns indexed by the fractures of H, and its entries satisfy:

$$M_H[\rho, \sigma] := \#\mathsf{Hom}((H \sharp \rho, c_\rho) \to (H \sharp \sigma, c_\sigma)) \,, \tag{4.1}$$

where c_ρ and c_σ are the canonical H-colourings of the fractured graphs $H \sharp \rho$ and $H \sharp \sigma$ (see Definition 2.1 and Observation 2.2). By ordering the columns and rows of M_H along a certain lattice, the following property was established in previous work.[14]

[13] Proofs of Lemma 4.2 and Lemma 4.3 can also be found in Section 3.1 in an earlier version [54] of [51].

[14] See Footnote 13.

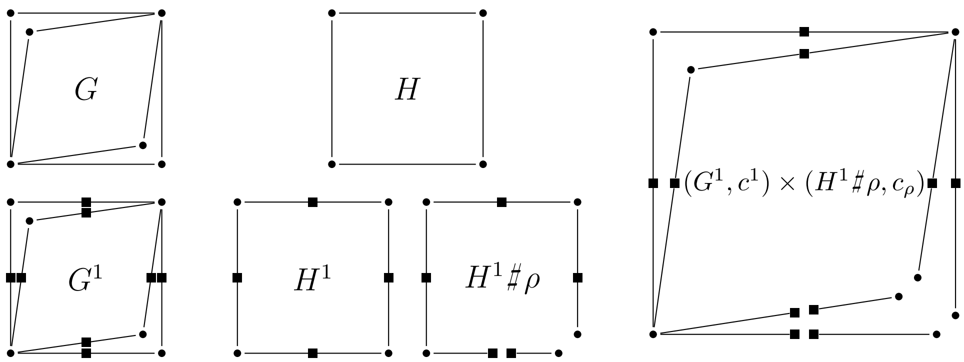

FIG. 3. *the tensor product of the H^1-coloured graphs (G^1, c^1) and $(H^1 \sharp \rho, c_\rho)$.*

LEMMA 4.3 ([51]). *For each graph H, the matrix M_H is nonsingular.*

If $\mathcal{G}$ is closed under *uncoloured* tensor products[15], then the hardness result can be achieved by applying the reduction of [18] verbatim. However, that reduction fails if $\mathcal{G}$ is not closed under uncoloured tensor products, and this closure property is very restrictive. Consider for example the class $\mathcal{G}$ of square-free graphs, i.e., graphs that do not contain the 4-cycle C_4 as a subgraph. Then $\mathcal{G}$ is clearly monotone and, since it contains the 3-subdivision of every graph, it is also somewhere dense by Lemma 2.9. However, $\mathcal{G}$ is not closed under (uncoloured) tensor products: the path on 2 edges P_2 is in $\mathcal{G}$, but $P_2 \times P_2 \notin \mathcal{G}$ since it contains a C_4.

The main insight of this section is a weakened closure property for monotone and somewhere dense graph classes, established in the lemma below. Combined with the characterisation of somewhere dense graph classes via r-subdivisions (Lemma 2.9), this property implies that any monotone and somewhere dense class is closed under tensor products of subdivisions of *coloured* graphs.

LEMMA 4.4. *Let $r \in \mathbb{N}_0$, let H be a graph without isolated vertices, and let (G, c) be an H-coloured graph on n vertices, and let ρ be a fracture of H^r. Then $(G^r, c^r) \times (H^r \sharp \rho, c_\rho)$ is a subgraph of the r-subdivision of a complete graph of order $O(kn)$, where $k = |E(H)|$ and the constants in the $O()$ notation depend only on r.*

Proof. Let $T = (G^r, c^r) \times (H^r \sharp \rho, c_\rho)$; see Figure 3 for an example. The claim follows from Claims 1, 2, and 3 below, with Claim 3 applied to $F = T$.

Claim 1. $|V(T)| = O(kn)$. Straightforward since G^r is a subgraph of K_n^r.

Claim 2: if x and y are distinct vertices of T of degree at least 3, then the length of any simple path from x to y is a multiple of $r + 1$.

To prove this, recall that T is H^r-coloured by $\hat{c}$ from Definition 4.1, and that $V(H^r)$ can be partitioned into $V(H)$ and a set S of kr fresh subdivision vertices. Let (u, v) be a vertex of T such that $\hat{c}(u, v) = s \notin V(H)$, that is, (u, v) is coloured with a subdivision vertex s. We show that (u, v) has degree at most 2 in T. Let s_1 and s_2 be the two neighbours of s in H^r. By the construction of (G^r, c^r), u has exactly two neighbours in G^r, say u_1 and u_2. Furthermore, $c^r(u_1) = s_1$ and $c^r(u_2) = s_2$. Since s has degree 2 in H^r, there are only two cases for ρ_s.

[15]The adjacency matrix of the tensor product of two uncoloured graphs G and F is the Kronecker product of the adjacency matrices of G and F.

- Case 1: $\rho_s = \{B\}$ where $B = \{\{s, s_1\}, \{s, s_2\}\}$. In this case s^B is the only vertex of $H\#\rho$ that is coloured by c_ρ with s. Since $\hat{c}(u, v) = s$ implies $c_\rho(v) = s$, we conclude that $v = s^B$. Hence (u, v) has exactly two neighbours in T, $(u_1, s_1^{B_1})$ and $(u_2, s_2^{B_2})$, where B_1 and B_2 are the blocks of ρ_{s_1} and ρ_{s_2} containing respectively $\{s, s_1\}$ and $\{s, s_2\}$.
- Case 2: $\rho_s = \{B, B'\}$ where $B = \{\{s, s_1\}\}$ and $B' = \{\{s, s_2\}\}$. In this case s^B and $s^{B'}$ are the only two vertices of $H\#\rho$ that are coloured by c_ρ with s. Since $\hat{c}(u, v) = s$ implies $c_\rho(v) = s$, we conclude that $v \in \{s^B, s^{B'}\}$. Assume that $v = S^B$; the other case is symmetric. Then the only neighbour of (u, v) in T is $(u_1, s_1^{B_1})$, where B_1 is the block of ρ_{s_1} that contains the edge $\{s, s_1\}$.

We conclude that the only vertices (u, v) of degree at least 3 in T satisfy $\hat{c}(u, v) \in V(H)$, implying that $c^r(u) \in V(H)$ and thus, by the definition of c^r, that $u \in V(G)$, hence u is not a subdivision vertex. The claim follows since the length of every simple path between two non-subdivision vertices u_1 and u_2 in G^r is a multiple of $(r + 1)$, and since T can be obtained from (G^r, c^r) by splitting vertices.

Claim 3: if F is a graph where the length of any simple path between two vertices of degree at least 3 is a multiple of $(r + 1)$, then F is a subgraph of the r-subdivision of a complete graph of order $O(|V(F)|)$.

Note first that we can deal with each connected component of F separately. Furthermore, the claim is clearly true if F is just a path (of any length). For what follows we can hence assume that F is connected and not isomorphic to a path. We say that a path P in F is *extendable* if its internal vertices have degree 2, one endpoint s_P (the "startpoint") has degree 1, and the other endpoint has degree at least 3. If P has length ℓ_P, then its *extension length* is the smallest $\ell'_P \in \mathbb{N}_0$ such that $\ell_P + \ell'_P$ is a multiple of $r+1$. Let F' be the graph formed from F by considering every extendable path P and adding a new length-ℓ'_P path from s_P (adding ℓ'_P fresh vertices to make up this path). Observe that, for every pair of non-isolated vertices u' and v' of F', if both u' and v' have degree not equal to 2, then the length of every simple path from u' to v' in F' is a multiple of $(r + 1)$. Therefore F' is a subgraph of the r-subdivision of a complete graph of order at most $O(|V(F')|) = O(|V(F)|)$, where the constants depend only on r. Moreover F is by construction a subgraph of F', which this concludes the proof of the claim. ∎

To establish the hardness of #Match($\mathcal{G}$), we first consider an edge-coloured version. Let G be a graph and $k \in \mathbb{N}$. A k-coloring of $E(G)$ is a map $c : E(G) \to \{1, \ldots, k\}$. A matching $M \subseteq E(G)$ is *edge-colorful* under if for every colour in $\{1, \ldots, k\}$ there is precisely one element of M with that colour.

Definition 4.5 (#ColMatch($\mathcal{G}$)). *Let $\mathcal{G}$ be a class of graphs. The problem #ColMatch($\mathcal{G}$) asks, on input $k \in \mathbb{N}$, a graph $G \in \mathcal{G}$, and a k-coloring c of $E(G)$, to compute the number of edge-colorful k-matchings in G. The problem is parameterised by k.*

Lemma 4.6. *Let $\mathcal{G}$ be a monotone somewhere dense class of graphs. Then the problem #ColMatch($\mathcal{G}$) is #W[1]-hard and, assuming ETH, cannot be solved in time $f(k) \cdot |G|^{o(k/\log k)}$ for any function f.*

Proof. Let $\mathcal{H}$ be a class of 3-regular expander graphs. Both the treewidth and the number of edges of the elements of $\mathcal{H}$ grow linearly in the number of vertices; that is, $|E(H)| \in \Theta(|V(H)|)$ and $\mathsf{tw}(H) \in \Theta(|V(H)|)$ for all $H \in \mathcal{H}$ (see, e.g., [37]). Hence theorems 2.14 and 2.16 imply that #cp-Hom($\mathcal{H} \to \mathcal{U}$) is #W[1]-hard and, assuming ETH, cannot be solved in time $f(|H|) \cdot |G|^{o(|H|/\log |H|)}$ for any function f. We will now show that #CP-HOM($\mathcal{H} \to \mathcal{U}$) $\leq^{\mathsf{FPT}}$ #COLMATCH($\mathcal{G}$).

Let $H \in \mathcal{H}$ and (G, c) be the input of #CP-HOM($\mathcal{H} \to \mathcal{U}$). By Lemma 2.9, there is $r \in \mathbb{N}_0$ such that $G^r \in \mathcal{G}$ for all $G \in \mathcal{U}$. Construct then H^r and (G^r, c^r), which clearly takes polynomial time. Let $k = |E(H^r)|$; clearly $k \in O(|H|)$ where the constants depend only on r. Now, by Lemma 3.1,

$$\#\mathsf{Hom}((H, \mathsf{id}_H) \to (G, c)) = \#\mathsf{Hom}((H^r, \mathsf{id}_{H^r}) \to (G^r, c^r))\,.$$

Next, we view surjectively H^r-coloured graphs $(\tilde{G}, \tilde{c})$ also as edge-coloured graphs where every edge $e = \{u, v\}$ is mapped to the colour $\{\tilde{c}(u), \tilde{c}(v)\}$. This allows us to invoke the results of [51] and deduce what follows.[16]

First, there is a unique function a from fractures of H^r to rationals such that, for every surjectively H^r-coloured graph $(\tilde{G}, \tilde{c})$, the number of edge-colourful k-matchings of $(\tilde{G}, \tilde{c})$ is:

$$\sum_{\rho} a(\rho) \cdot \#\mathsf{Hom}((H^r \sharp \rho, c_\rho) \to (\tilde{G}, \tilde{c}))\,, \tag{4.2}$$

where the sum is over all fractures of H^r. Additionally, a satisfies:

$$a(\top) = \prod_{v \in V(H^r)} (-1)^{\mathsf{deg}(v)-1} \cdot (\mathsf{deg}(v) - 1)!\,, \tag{4.3}$$

where $\top$ is the coarsest fracture, that is, for each $v \in V(H^r)$ the partition $\top_v$ only contains a singleton block (and therefore $H^r \sharp \top = H^r$). In particular, it is easy to see that

$$a(\top) = \pm 2^{|V(H)|} \neq 0\,.$$

Now let σ be a fracture of H^r. Considering (4.2) with $(\tilde{G}, \tilde{c}) = (G^r, c^r) \times (H^r \sharp \sigma, c_\sigma)$ and applying Lemma 4.2, the number of colorful k-matchings in $(G^r, c^r) \times (H^r \sharp \sigma, c_\sigma)$ equals:

$$\sum_{\rho} a(\rho) \cdot \#\mathsf{Hom}((H^r \sharp \rho, c_\rho) \to (G^r, c^r)) \cdot \#\mathsf{Hom}((H^r \sharp \rho, c_\rho) \to (H^r \sharp \sigma, c_\sigma)) \tag{4.4}$$

By Lemma 4.4, $(G^r, c^r) \times (H^r \sharp \sigma, c_\sigma)$ is a subgraph of the r-subdivision of a complete graph, which is in $\mathcal{G}$ by our choice of r. Since $\mathcal{G}$ is monotone this implies $(G^r, c^r) \times (H^r \sharp \sigma, c_\sigma) \in \mathcal{G}$, too. Hence, if we have an oracle for #COLMATCH($\mathcal{G}$), then we can compute the value of (4.4), while $\#\mathsf{Hom}((H^r \sharp \rho, c_\rho) \to (H^r \sharp \sigma, c_\sigma))$ can obviously be computed in a time that is a function of $|H|$ and r. Thus, by letting $\mathsf{coeff}(\rho) := a(\rho) \cdot \#\mathsf{Hom}((H^r \sharp \rho, c_\rho) \to (G^r, c^r))$, in FPT time we obtain a system of linear equations with unknowns $\mathsf{coeff}(\rho)$ and whose matrix is M_{H^r}, see (4.1). By Lemma 4.3 M_{H^r} is nonsingular, hence by solving the system we can retrieve:

$$\mathsf{coeff}(\top) = a(\top) \cdot \#\mathsf{Hom}((H^r \sharp \top, c_\top) \to (G^r, c^r)) = a(\top) \cdot \#\mathsf{Hom}((H^r, \mathsf{id}_{H^r}) \to (G^r, c^r))\,.$$

Since $a(\top) \neq 0$, we can divide by $a(\top)$ and recover $\#\mathsf{Hom}((H^r, \mathsf{id}_{H^r}) \to (G^r, c^r))$ as desired. This concludes the parameterized reduction to #COLMATCH($\mathcal{G}$) and proves the thesis. ∎

[16] In [51], the number of edge-colourful k-matchings of G is denoted by $\#\mathsf{ColEdgeSub}(\Phi, k \to G)$, where Φ is the graph property of being a matching. The identities (4.2) and (4.3) are immediate consequences of Lemma 4.1 and Corollary 4.3 in [51] (see also Lemma 3.1 and Corollary 3.3 in an earlier version [54] of [51]).

With the hardness results for $\#\textsc{ColMatch}(\mathcal{G})$ above, we can finally obtain our complexity dichotomy for $\#\textsc{Match}(\mathcal{G})$. First, we prove:

THEOREM 4.7. *Let $\mathcal{G}$ be a monotone somewhere dense class of graphs. Then $\#\textsc{Match}(\mathcal{G})$ is $\#\mathsf{W}[1]$-hard and, assuming ETH, cannot be solved in time $f(k) \cdot |G|^{o(k/\log k)}$ for any function f.*

Proof. A well-known application of inclusion-exclusion (see, e.g., [16, Lemma 1.34]) yields a parameterized reduction from $\#\textsc{ColMatch}(\mathcal{G})$ to $\#\textsc{Match}(\mathcal{G}')$ that preserves the parameter, where $\mathcal{G}'$ is the class of all subgraphs of $\mathcal{G}$. By monotonicity $\mathcal{G}' = \mathcal{G}$, so the claim of Lemma 4.6 holds for $\#\textsc{Match}(\mathcal{G})$, too. ∎

Finally, we obtain:

COROLLARY 4.8 (Theorem 1.2, restated). *Let $\mathcal{G}$ be a monotone class of graphs and assume that ETH holds. Then $\#\textsc{Match}(\mathcal{G})$ is fixed-parameter tractable if and only if $\mathcal{G}$ is nowhere dense. In particular, 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 $\#\textsc{Match}(\mathcal{G})$ cannot be solved in time $f(k) \cdot |G|^{o(k/\log k)}$ for any function f.*

Proof. Immediate from Theorem 2.11 and Theorem 4.7. ∎

REMARK 4.9. *Unless $\#\mathrm{P} = \mathrm{P}$, Corollary 4.8 / Theorem 1.2 cannot be strengthened to achieve polynomial time tractability of $\#\textsc{Match}(\mathcal{G})$ for nowhere dense and monotone $\mathcal{G}$. Let indeed $\mathcal{G}$ be the class of all K_8-minor-free graphs. Then $\mathcal{G}$ is clearly monotone, and since it does not contain the subdivisions of cliques larger than 7, it is also nowhere dense by Lemma 2.9. However, as shown recently by Curticapean and Xia [21], counting perfect matchings (i.e., k-matchings with $k = n/2$) in K_8-minor-free graphs is $\#\mathrm{P}$-hard.*

4.2. Counting Subgraphs: Proofs of Theorems 1.3 and 1.4. Equipped with our hardness results for counting k-matchings, we move towards proving hardness for counting subgraphs.

THEOREM 4.10 (Theorem 1.3, restated). *Let $\mathcal{H}$ and $\mathcal{G}$ be graph classes such that $\mathcal{H}$ is hereditary and $\mathcal{G}$ is monotone. Then Table 3 exhaustively classifies the complexity of $\#\textsc{Sub}(\mathcal{H} \to \mathcal{G})$.*

Proof. Let us first show that the cases for $\mathcal{H}$ and $\mathcal{G}$ in Table 3 are exhaustive and mutually exclusive. For $\mathcal{G}$ this is straightforward. For $\mathcal{H}$, the first row and the rest are mutually exclusive and exhaustive, since rows 2, 3 and 4 all imply $\mathrm{m}(\mathcal{H}) = \infty$. To see that rows 2, 3, and 4 are mutually exclusive and exhaustive for $\mathrm{m}(\mathcal{H}) = \infty$, note that in that case Theorem 2.5 implies that at least one of $\mathrm{m}_{\mathsf{ind}}(\mathcal{H})$, $\beta_{\mathsf{ind}}(\mathcal{H})$ and $\omega(\mathcal{H})$ is unbounded.

Let us now prove the entries of Table 3. The first row is due to Curticapean and Marx [20], and the FPT result in the first column follows from Theorem 2.11. The intractability results in the second row follow from Theorem 4.7 and the fact that $\mathrm{m}_{\mathsf{ind}}(\mathcal{H}) = \infty$ implies that $\mathcal{H}$ contains all matchings (since $\mathcal{H}$ is hereditary). For the second column, note that $\omega(\mathcal{G}) = \infty$ and $\mathcal{G}$ being monotone implies that $\mathcal{G} = \mathcal{U}$; the dichotomy of Curticapean and Marx [20] then applies again.[17] Next, we prove the

[17]The tight conditional lower bounds in the second column follow from the fact that the respective entries subsume counting k-cliques in arbitrary graphs, and counting k-by-k bicliques in bipartite graphs. The tight bound of the former was shown in [12, 13], and the tight bound of the latter was implicitly shown in [20], and explicitly in [26]; while [26] studies *induced* subgraphs in bipartite graphs, we note that all bicliques in a bipartite graph must be induced.

	$\mathcal{G}$ 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
$\mathrm{m}_{\mathsf{ind}}(\mathcal{H}) < \infty$ $\beta_{\mathsf{ind}}(\mathcal{H}) < \infty$ $\omega(\mathcal{H}) = \infty$	P	hard†	P	P

TABLE 3

The complexity of $\#\text{SUB}(\mathcal{H} \to \mathcal{G})$ *for hereditary* $\mathcal{H}$ *and monotone* $\mathcal{G}$ *(Theorem 1.3). P and FPT stand respectively for polynomial-time tractability and fixed-parameter tractability, hard means* #W[1]*-hard and without an algorithm running in time* $f(|H|) \cdot |G|^{o(|V(H)|/\log|V(H)|)}$ *for any function* f *unless ETH fails, and hard*† *means the same but with a lower bound of* $f(|H|) \cdot |G|^{o(|V(H)|)}$*. The FPT entry cannot be strengthened to P unless* P = #P*, see Remark 4.12.*

remaining entries.

• Row 3, Column 3: if $\beta(\mathcal{G}) = \beta_{\mathsf{ind}}(\mathcal{H}) = \infty$ then $\#\text{SUB}(\mathcal{H} \to \mathcal{G})$ is hard. Since $\mathcal{H}$ is hereditary, it contains all bicliques. Since $\mathcal{G}$ is monotone, it contains all bipartite graphs. Hence $\#\text{SUB}(\mathcal{H} \to \mathcal{G})$ is at least as hard as counting k-by-k bicliques in bipartite graphs, which is known to be hard [20].[18]

• Row 3, Column 4: if $\mathrm{m}_{\mathsf{ind}}(\mathcal{H}), \omega(\mathcal{G}), \beta(\mathcal{G}) < \infty$ then $\#\text{SUB}(\mathcal{H} \to \mathcal{G})$ is in polynomial time. Let (H, G) be the input of $\#\text{SUB}(\mathcal{H} \to \mathcal{G})$. If $\omega(H) > \omega(\mathcal{G})$ or $\beta_{\mathsf{ind}}(H) > \beta(\mathcal{G})$, then we can output 0. We can thus restrict the problem to those H such that $\omega(H) \le \omega(\mathcal{G})$ and $\beta_{\mathsf{ind}}(H) \le \beta(\mathcal{G})$. Recall that $\mathrm{m}_{\mathsf{ind}}(H) \le \mathrm{m}_{\mathsf{ind}}(\mathcal{H}) < \infty$. By the contrapositive of Theorem 2.5, there is a monotonically increasing function R such that:

$$\mathrm{m}(H) \le R(\mathrm{m}_{\mathsf{ind}}(H), \omega(H), \beta_{\mathsf{ind}}(H)) \le R(\mathrm{m}_{\mathsf{ind}}(\mathcal{H}), \omega(\mathcal{G}), \beta(\mathcal{G})) < \infty,$$

where the second inequality holds by monotonicity of R and the third one by the boundedness of all three arguments. We therefore obtain polynomial time as in the first row.

• Row 4, Columns 3 and 4: if $\mathrm{m}_{\mathsf{ind}}(\mathcal{H}), \beta_{\mathsf{ind}}(\mathcal{H}), \omega(\mathcal{G}) < \infty$, then $\#\text{SUB}(\mathcal{H} \to \mathcal{G})$ is in polynomial time. Let (H, G) be the input of $\#\text{SUB}(\mathcal{H} \to \mathcal{G})$. If $\omega(H) > \omega(\mathcal{G})$ then we output 0, hence we can assume that $\omega(H) \le \omega(\mathcal{G})$. Similarly to the previous case, we then obtain polynomial time since

$$\mathrm{m}(H) \le R(\mathrm{m}_{\mathsf{ind}}(H), \omega(H), \beta_{\mathsf{ind}}(H)) \le R(\mathrm{m}_{\mathsf{ind}}(H), \omega(\mathcal{G}), \beta_{\mathsf{ind}}(H)) < \infty.$$

• Rows 3 and 4, Column 1: $\#\text{SUB}(\mathcal{H} \to \mathcal{G})$ is in polynomial time. We show that $\omega(\mathcal{G}), \beta(\mathcal{G}) < \infty$; then the same arguments used for Rows 3 and 4 of Column 4

[18] See Footnote 17 for the tight conditional lower bound.

apply. Suppose by contradiction that $\max(\omega(\mathcal{G}), \beta(\mathcal{G})) = \infty$. Since $\mathcal{G}$ is monotone, if $\omega(\mathcal{G}) = \infty$ then $\mathcal{G}$ contains (the 0-subdivision of) every clique, and if $\beta(\mathcal{G}) = \infty$ then $\mathcal{G}$ contains all bipartite graphs and thus the 1-subdivision of every clique. In any case Lemma 2.9 implies that $\mathcal{G}$ is somewhere dense, contradicting the assumptions. ∎

Theorem 1.4 follows immediately.

COROLLARY 4.11 (Theorem 1.4, restated). *Let $\mathcal{H}$ and $\mathcal{G}$ be monotone graph classes 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.*

Proof. If $\mathcal{H}$ is monotone then $\mathcal{H}$ is hereditary and Theorem 1.3 applies. The union of the first row and the first column of Table 3 yield the tractable case; the union of the remaining entries yield the intractable case and the lower bounds. ∎

We conclude this section with a remark.

REMARK 4.12. *Let $\mathcal{H}$ and $\mathcal{G}$ be the classes of graphs of degrees bounded by 2 and 3, respectively. Then $\#\text{SUB}(\mathcal{H} \to \mathcal{G})$ subsumes the $\#\mathrm{P}$-hard problem of counting Hamiltonian cycles in 3-regular graphs. Since both classes are monotone (and thus also hereditary), since $\mathrm{m}_{\mathsf{ind}}(\mathcal{H}) = \infty$, and since classes of bounded degree graphs are nowhere dense (see e.g. [36]), this shows that the FPT entry in Table 3 cannot be strengthened to P unless $\#\mathrm{P} = \mathrm{P}$.*

5. Counting Induced Subgraphs.

This section is devoted to the proofs of Theorem 1.5 and Theorem 1.6. We begin in Section 5.1 by analysing the problem of counting independent sets and proving Theorem 1.5; this is the most technical part. We then prove Theorem 1.6 in Section 5.2.

5.1. Counting Independent Sets: Proof of Theorem 1.5.

Given a class of graphs $\mathcal{G}$, the problem $\#\text{INDSET}(\mathcal{G})$ asks, on input $k \in \mathbb{N}$ and a graph $G \in \mathcal{G}$, to compute the number of independent sets of size k (also called *k-independent sets*) in G. In this section we prove hardness results for $\#\text{INDSET}(\mathcal{G})$ and leverage them to $\#\text{INDSUB}(\mathcal{H} \to \mathcal{G})$. To this end we will rely on subgraphs induced by sets of edges; they play a role similar to that of fractured graphs in Section 4. Given a graph F and a set $A \subseteq E(F)$, we denote the subgraph $(V(F), A)$ by $F[A]$. For what follows observe that, for any $A \subseteq E(F)$, the identity function on $V(F)$, which we denote by id_F, is a surjective F-colouring of $F[A]$. Now recall Definition 4.1. We start with the following simple variation of Lemma 4.4.

LEMMA 5.1. *Let $r \in \mathbb{N}_0$, let H be a graph without isolated vertices, let G be an H-coloured graph, and let $A \subseteq E(H^r)$. Then $(G^r, c^r) \times (H^r[A], \mathsf{id}_{H^r})$ is a subgraph of $K^r_{|V(G)|}$.*

Proof. Let $n = |V(G)|$. First, note that $(G^r, c^r) \times (H^r, \mathsf{id}_{H^r}) = (G^r, c^r)$, and by construction (G^r, c^r) is a subgraph of K^r_n. Next, for every $A \subseteq E(H^r)$ the graph $(G^r, c^r) \times (H^r[A], \mathsf{id}_{H^r})$ is obtained from (G^r, c^r) by deleting edges — specifically, for every $e = \{u, v\} \in E(H^r) \setminus A$, delete from G^r all edges between vertices coloured with u and vertices coloured with v. Thus $(G^r, c^r) \times (H^r[A], \mathsf{id}_{H^r})$ is a subgraph of K^r_n too. ∎

Recall that $\#\text{COLMATCH}(\mathcal{G})$, the problem of counting edge-colourful k-matchings, was the key subproblem in the hardness proofs for $\#\text{SUB}(\mathcal{H} \to \mathcal{G})$ — see Section 4.1. In the case of $\#\text{INDSUB}(\mathcal{H} \to \mathcal{G})$, the key subproblem turns out to be that of counting vertex-colourful independent sets. Let G be a graph and let $c : V(G) \to \{1, \ldots, k\}$ be a coloring of $V(G)$. A set $U \subseteq V(G)$ is *vertex-colorful* if for every colour in $\{1, \ldots, k\}$ there is precisely one element of U with that colour.

DEFINITION 5.2 ($\#\textsc{ColIndSet}(\mathcal{G})$). *Let $\mathcal{G}$ be a class of graphs. The problem $\#\textsc{ColIndSet}(\mathcal{G})$ asks, on input $k \in \mathbb{N}$, a graph $G \in \mathcal{G}$, and a k-coloring c of $V(G)$, to compute the number of vertex-colorful k-independent sets in G. The problem is parameterised by k.*

Our goal is to show that $\#\textsc{ColIndSet}(\mathcal{G})$ is intractable whenever $\mathcal{G}$ is monotone and somewhere dense. As for $\#\textsc{ColMatch}(\mathcal{G})$ in Section 4.1, the reduction relies on solving a system of linear equations. Let H be a graph. The square matrix N_H has its rows and columns indexed by the subsets of $E[H]$, and its entries satisfy

$$N_H[A, B] = \#\mathsf{Hom}((H[A], \mathsf{id}_H) \to (H[B], \mathsf{id}_H))\,. \tag{5.1}$$

Similarly to the matrix M_H in Section 4.1, the following was established in prior work:

LEMMA 5.3 ([26]). *For each graph H, the matrix N_H is nonsingular.*

We are now able to establish intractability of $\#\textsc{ColIndSet}(\mathcal{G})$.

LEMMA 5.4. *Let $\mathcal{G}$ be a monotone somewhere dense class of graphs. Then the problem $\#\textsc{ColIndSet}(\mathcal{G})$ is $\#\mathsf{W}[1]$-complete and, assuming ETH, cannot be solved in time $f(k) \cdot |G|^{o(k/\log k)}$ for any function f.*

Proof. The proof is similar to that of Lemma 4.6. First, since $\mathcal{G}$ is monotone and somewhere dense, by Lemma 2.9 there exists $r \in \mathbb{N}_0$ such that $G^r \in \mathcal{G}$ for every $G \in \mathcal{U}$. Second, let $\mathcal{H}$ be a class of 3-regular expander graphs. By theorems 2.14 and 2.16, $\#\textsc{cp-Hom}(\mathcal{H} \to \mathcal{U})$ is $\#$W[1]-hard and assuming ETH cannot be solved in time $f(|H|) \cdot |G|^{o(|H|/\log |H|)}$ for any function f. We show a parameterized reduction from $\#\textsc{cp-Hom}(\mathcal{H} \to \mathcal{U})$ to $\#\textsc{ColIndSet}(\mathcal{G})$.

Let $(H, (G, c))$ be the input to $\#\textsc{cp-Hom}(\mathcal{H} \to \mathcal{U})$. Our reduction starts by constructing H^r and (G^r, c^r), which by Lemma 3.1 satisfy

$$\#\mathsf{Hom}((H, \mathsf{id}_H) \to (G, c)) = \#\mathsf{Hom}((H^r, \mathsf{id}_{H^r}) \to (G^r, c^r))\,.$$

Let $k = |V(H^r)|$; clearly $k \in O(|H|)$ since r is a constant independent of H. Our goal is to use the oracle for $\#\textsc{ColIndSet}(\mathcal{G})$ to compute $\#\mathsf{Hom}((H^r, \mathsf{id}_{H^r}) \to (G^r, c^r))$. From now on we view surjectively H^r-coloured graphs $(\tilde{G}, \tilde{c})$ also as vertex-coloured graphs with colouring $\tilde{c}$. This allows us to invoke [26, Lemma 8] and obtain what follows.[19]

First, there is a unique function $\hat{a}$ from subsets of $E[H^r]$ to rationals such that, for every surjectively H^r-coloured graph $(\tilde{G}, \tilde{c})$, the number of vertex-colourful k-independent sets in $(\tilde{G}, \tilde{c})$ equals

$$\sum_A \hat{a}(A) \cdot \#\mathsf{Hom}((H^r[A], \mathsf{id}_{H^r}) \to (\tilde{G}, \tilde{c}))\,, \tag{5.2}$$

where the sum is over all subsets of $E[H^r]$. Additionally,

$$\hat{a}(E(H^r)) = \pm\hat{\chi} \neq 0\,, \tag{5.3}$$

[19] In [26] the number of colourful k-independent sets in a surjectively H^r-coloured graph $\hat{G}$ is denoted by $\#\mathsf{cp\text{-}IndSub}(\Phi \to_{H^r} \hat{G})$, where Φ is the graph property of being an independent set.

where $\hat{\chi}$ is the so-called alternating enumerator for the graph property of being an independent set — we omit the definition since the only property needed for $\hat{\chi}$ is it being easily computable and non-zero (see [26]).

Now consider (5.2) with $(\tilde{G}, \tilde{c}) = (G^r, c^r) \times (H^r[B], \mathsf{id}_{H^r})$ and apply Lemma 4.2. We deduce that the number of vertex-colourful k-independent sets in $(\tilde{G}, \tilde{c})$ is

$$\sum_A \hat{a}(A) \cdot \#\mathsf{Hom}((H^r[A], \mathsf{id}_{H^r}) \to (G^r, c^r)) \cdot \#\mathsf{Hom}((H^r[A], \mathsf{id}_{H^r}) \to (H^r[B], \mathsf{id}_{H^r})).$$

By Lemma 5.1, for every $B \subseteq E(H^r)$ of H^r the graph $(G^r, c^r) \times (H^r[B], \mathsf{id}_{H^r})$ is a subgraph of the r-subdivision of a complete graph; by the monotonicity of $\mathcal{G}$ and by the choice of r this implies $(G^r, c^r) \times (H^r[B], \mathsf{id}_{H^r}) \in \mathcal{G}$, see Lemma 2.9. Thus, as in the proof of Lemma 4.6, by using an oracle for #ColIndSet($\mathcal{G}$) we can construct in FPT time a system of linear equations whose matrix N_{H^r} is nonsingular by Lemma 5.3. Since $\hat{a}(E(H^r)) \neq 0$ by (5.3), solving this system enables us to compute

$$\#\mathsf{Hom}((H^r[E(H^r)], \mathsf{id}_{H^r}) \to (G^r, c^r)) = \#\mathsf{Hom}((H^r, \mathsf{id}_{H^r}) \to (G^r, c^r)),$$

concluding the proof. ∎

With the above hardness results for #ColIndSet($\mathcal{G}$), we can finally prove complexity dichotomies for its non-coloured counterpart #IndSet($\mathcal{G}$). We start by porting Lemma 5.4 from #ColIndSet($\mathcal{G}$) to #IndSet($\mathcal{G}$).

Theorem 5.5. *Let $\mathcal{G}$ be a monotone somewhere dense class of graphs. Then* #IndSet($\mathcal{G}$) *is* #W[1]*-hard and, assuming ETH, cannot be solved in time $f(k) \cdot |G|^{o(k/\log k)}$ for any function f.*

Proof. Almost identical to the proof of Theorem 4.7: when $\mathcal{G}$ is monotone, #ColIndSet($\mathcal{G}$) can be reduced in FPT time to #IndSet($\mathcal{G}$) via inclusion-exclusion while preserving the parameter (see, for instance, [16, Lemma 1.34]), and the claim then follows by Lemma 5.4. ∎

We can finally prove Theorem 1.5 as a simple corollary.

Corollary 5.6 (Theorem 1.5, restated). *Let $\mathcal{G}$ be a monotone class of graphs and assume that ETH holds. Then* #IndSet($\mathcal{G}$) *is fixed-parameter tractable if and only if $\mathcal{G}$ is nowhere dense. In particular, if $\mathcal{G}$ is nowhere dense then* #IndSet($\mathcal{G}$) *can be solved in time $f(k) \cdot |V(G)|^{1+o(1)}$ for some computable function f; otherwise* #IndSet($\mathcal{G}$) *cannot be solved in time $f(k) \cdot |G|^{o(k/\log k)}$ for any function f.*

Proof. Immediate by Theorem 2.11 and Theorem 5.5. ∎

We conclude with a remark.

Remark 5.7. *Corollary 5.6 cannot be strengthened to polynomial-time tractability of* #IndSet($\mathcal{G}$) *when $\mathcal{G}$ is nowhere dense and monotone, unless #P = P: graphs of degree at most 3 form such a class, yet counting independent sets in them is #P-hard [35].*

5.2. Counting Induced Subgraphs: Proof of Theorem 1.6. Equipped with our complexity dichotomy for #IndSet($\mathcal{G}$), we can now prove our complexity dichotomies for #IndSub($\mathcal{H} \to \mathcal{G}$). First, we consider the case that $\mathcal{H}$ is monotone.

Corollary 5.8. *Let $\mathcal{H}$ and $\mathcal{G}$ be monotone graph classes and assume that ETH holds. Then* #IndSub($\mathcal{H} \to \mathcal{G}$) *is fixed-parameter tractable if $\mathcal{H}$ is finite or $\mathcal{G}$ is nowhere dense; otherwise* #IndSub($\mathcal{H} \to \mathcal{G}$) *is* #W[1]*-complete and cannot be solved in time $f(|H|) \cdot |G|^{o(|V(H)|/\log(|V(H)|))}$ for any function f.*

	$\mathcal{G}$ nowhere dense	$\mathcal{G}$ somewhere dense $\omega(\mathcal{G}) = \infty$	$\mathcal{G}$ somewhere dense $\omega(\mathcal{G}) < \infty$ $\alpha(\mathcal{G}) = \infty$
$\mathcal{H}$ finite	P	P	P
$\alpha(\mathcal{H}) = \infty$	FPT	#W[1]-hard not in $f(k) \cdot n^{o(k)}$	#W[1]-hard not in $f(k) \cdot n^{o(k/\log k)}$
$\alpha(\mathcal{H}) < \infty$ $\omega(\mathcal{H}) = \infty$	P	#W[1]-hard not in $f(k) \cdot n^{o(k)}$	P

TABLE 4

The complexity of #INDSUB($\mathcal{H} \to \mathcal{G}$) *for hereditary* $\mathcal{H}$ *and monotone* $\mathcal{G}$. *P and FPT stand respectively for polynomial-time tractability and fixed-parameter tractability, and hard means* #W[1]*-hard and without an algorithm running in time* $f(k) \cdot n^{o(k/\log(k))}$ *for any function* f *unless ETH fails, where* $k = |V(H)|$ *and* $n = |V(G)|$. *The FPT entry cannot be strengthened to P unless* P = #P, *see Remark 5.7.*

Proof. If $\mathcal{H}$ is finite then #INDSUB($\mathcal{H} \to \mathcal{G}$) is clearly in polynomial time (and thus fixed-parameter tractable) since the brute-force algorithm runs in time $O(|G|^{|H|})$. If $\mathcal{G}$ is nowhere dense then the fixed-parameter tractability follows by Theorem 2.11. Finally, if $\mathcal{H}$ is monotone and infinite then it contains all independent sets, and thus #INDSUB($\mathcal{H} \to \mathcal{G}$) subsumes #INDSET($\mathcal{G}$); in which case Theorem 5.5 yields the lower bound for somewhere dense $\mathcal{G}$. ∎

Next, we consider the case that $\mathcal{H}$ is hereditary. We obtain a refined complexity classification that subsumes the one of Corollary 5.8 and yields Theorem 1.6.

THEOREM 5.9 (Theorem 1.6, restated). *Let* $\mathcal{H}$ *and* $\mathcal{G}$ *be graph classes such that* $\mathcal{H}$ *is hereditary and* $\mathcal{G}$ *is monotone. Then Table 4 exhaustively classifies the complexity of* #INDSUB($\mathcal{H} \to \mathcal{G}$).

Proof. The cases for $\mathcal{G}$ and $\mathcal{H}$ in Table 4 are mutually exclusive and exhaustive by Ramsey's Theorem (Theorem 2.4). Let us then prove the entries of Table 4.

The first row holds since for finite $\mathcal{H}$ the brute-force algorithm runs in polynomial time, and the FPT result follows from Theorem 2.11. For the intractability results in the second column, note that since $\mathcal{G}$ is monotone and infinite then $\mathcal{G} = \mathcal{U}$, and since $\mathcal{H}$ is hereditary, the cases $\alpha(\mathcal{H}) = \infty$ and $\omega(\mathcal{H}) = \infty$ subsume respectively #INDSET($\mathcal{U}$) and #CLIQUE($\mathcal{U}$). Both are canonical #W[1]-hard problems and cannot be solved in time $f(k) \cdot n^{o(k)}$ unless ETH fails [12, 13].[20] The intractability results in the third column follows from Theorem 5.5 since $\mathcal{H}$ being hereditary and $\alpha(\mathcal{H}) = \infty$ imply that #INDSUB($\mathcal{H} \to \mathcal{G}$) subsumes #INDSET($\mathcal{G}$).

It remains to prove the first and the third entry of the third row. Note that both entries assume $\omega(\mathcal{G}) < \infty$ and $\alpha(\mathcal{H}) < \infty$. Let then (H, G) be the input. If $\omega(H) > \omega(\mathcal{G})$ then we can immediately return 0. Otherwise $|V(H)| \le R(\omega(\mathcal{G}), \alpha(\mathcal{H})) < \infty$, where R is Ramsey's function (see Theorem 2.4), and the brute-force algorithm runs in polynomial time. ∎

[20]The lower bound in [12, 13] applies to counting k-cliques, and we note that counting k-cliques and counting k-independent sets are interreducible by taking the complement of the host.

6. Outlook. Due to the absence of a general dichotomy [56], 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 [31, 33, 34] for recent work on specific hereditary hosts and [32, 7] 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 [20] and degenerate graphs [9].

REFERENCES

[1] Hans Adler and Isolde Adler. Interpreting nowhere dense graph classes as a classical notion of model theory. *European J. Combin.*, 36:322–330, 2014. doi:10.1016/j.ejc.2013.06.048.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Marco Bressan, Leslie Ann Goldberg, Kitty Meeks, and Marc Roth. Counting subgraphs in somewhere dense graphs. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 27:1–27:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. URL: https://doi.org/10.4230/LIPIcs.ITCS.2023.27, doi:10.4230/LIPICS.ITCS.2023.27.

[9] 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.

[10] 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.

[11] 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.

[12] Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David W. Juedes, Iyad A. Kanj, and Ge Xia. Tight lower bounds for certain parameterized NP-hard problems. *Inform. and Comput.*, 201(2):216–231, 2005. doi:10.1016/j.ic.2005.05.001.

[13] Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. System Sci.*, 72(8):1346–1367, 2006. doi:10.1016/j.jcss.2006.04.007.

[14] 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.

[15] 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.

[16] Radu Curticapean. *The simple, little and slow things count: On parameterized counting complexity*. PhD thesis, Saarland University, 2015. URL: http://scidok.sulb.uni-saarland.de/volltexte/2015/6217/.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] Radu Curticapean and Mingji Xia. Parameterizing the permanent: Hardness for fixed excluded minors. In *Proceedings of the 5th Symposium on Simplicity in Algorithms, (SOSA@SODA)*, pages 297–307. SIAM, 2022. doi:10.1137/1.9781611977066.23.

[22] 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.

[23] 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.

[24] Josep Díaz, Maria J. Serna, and Dimitrios M. Thilikos. Counting H-colorings of partial k-trees. *Theoret. Comput. Sci.*, 281(1-2):291–309, 2002. doi:10.1016/S0304-3975(02)00017-8.

[25] Reinhard Diestel. *Graph Theory (5th Edition)*. Springer, 2017. doi:10.1007/978-3-662-53622-3.

[26] 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.

[27] 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.

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

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] Catherine S. Greenhill. The complexity of counting colourings and independent sets in sparse graphs and hypergraphs. *Comput. Complexity*, 9(1):52–72, 2000. doi:10.1007/PL00001601.

[36] 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.

[37] Martin Grohe and Dániel Marx. On tree width, bramble size, and expansion. *J. Combin. Theory Ser. B*, 99(1):218–228, 2009. doi:10.1016/j.jctb.2008.06.004.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] László Lovász. *Large Networks and Graph Limits*, volume 60 of *Colloquium Publications*. Amer-

ican Mathematical Society, 2012. URL: http://www.ams.org/bookstore-getitem/item=COLL-60.

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

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

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] Neil Robertson and Paul D. Seymour. Graph minors. V. Excluding a planar graph. *J. Combin. Theory Ser. B*, 41(1):92–114, 1986. doi:10.1016/0095-8956(86)90030-4.

[53] Marc Roth. *Counting Problems on Quantum Graphs: Parameterized and Exact Complexity Classifications*. PhD thesis, Saarland University, 2019. URL: https://scidok.sulb.uni-saarland.de/bitstream/20.500.11880/27575/1/thesis.pdf.

[54] Marc Roth, Johannes Schmitt, and Philip Wellnitz. Detecting and counting small subgraphs, and evaluating a parameterized tutte polynomial: Lower bounds via toroidal grids and cayley graph expanders. *CoRR*, abs/2011.03433, 2020. arXiv:2011.03433.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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)*, page 1451–1460, 2017. doi:10.1145/3038912.3052653.

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