Characterizing the easy-to-find subgraphs from the viewpoint of polynomial-time algorithms, kernels, and Turing kernels *

Bart M. P. Jansen[†] Dániel Marx [‡]

Abstract

We study two fundamental problems related to finding subgraphs: (1) given graphs G and H, Subgraph Test asks if H is isomorphic to a subgraph of G, (2) given graphs G, H, and an integer t, Packing asks if G contains t vertex-disjoint subgraphs isomorphic to H. For every graph class $\mathcal{F}$, let $\mathcal{F}$-Subgraph Test and $\mathcal{F}$-Packing be the special cases of the two problems where H is restricted to be in $\mathcal{F}$. Our goal is to study which classes $\mathcal{F}$ make the two problems tractable in one of the following senses:

- (randomized) polynomial-time solvable,
- admits a polynomial (many-one) kernel (that is, has a polynomial-time preprocessing procedure that creates an equivalent instance whose size is polynomially bounded by the size of the solution), or
- admits a polynomial Turing kernel (that is, has an adaptive polynomial-time procedure that reduces the problem to a polynomial number of instances, each of which has size bounded polynomially by the size of the solution).

To obtain a more robust setting, we restrict our attention to hereditary classes $\mathcal{F}$.

It is known that if every component of every graph in $\mathcal{F}$ has at most two vertices, then $\mathcal{F}$-Packing is polynomial-time solvable, and NP-hard otherwise. We identify a simple combinatorial property (every component of every graph in $\mathcal{F}$ either has bounded size or is a bipartite graph with one of the sides having bounded size) such that if a hereditary class $\mathcal{F}$ has this property, then $\mathcal{F}$-Packing admits a polynomial kernel, and has no polynomial (many-one) kernel otherwise, unless the polynomial hierarchy collapses. Furthermore, if $\mathcal{F}$ does not have this property, then $\mathcal{F}$-Packing is either WK[1]-hard, W[1]-hard, or Long Path-hard, giving evidence that it does not admit polynomial Turing kernels either.

For $\mathcal{F}$-Subgraph Test, we show that if every graph of a hereditary class $\mathcal{F}$ satisfies the property that it is possible to delete a bounded number of vertices such that every remaining component has size at most two, then $\mathcal{F}$-Subgraph Test is solvable in randomized polynomial time and it is NP-hard otherwise. We introduce a combinatorial property called (a,b,c,d)-splittability and show that if every graph in a hereditary class $\mathcal{F}$ has this property, then $\mathcal{F}$-Subgraph Test admits a polynomial Turing kernel and it is WK[1]-hard, W[1]-hard, or Long Path-hard otherwise. We do not give a complete characterization of the cases when $\mathcal{F}$-Subgraph Test admits polynomial many-one kernels, but show examples that this question is much more fragile than the characterization for Turing kernels.

1 Introduction

Many classical algorithmic problems on graphs can be defined in terms of finding a subgraph that is isomorphic to a certain pattern graph. For example, the polynomial-time solvable problem of finding perfect matchings and the NP-hard Hamiltonian Cycle and Clique problems arise this way. The goal of the paper is to understand which pattern graphs make this problem easy with respect to polynomial-time solvability and polynomial-time preprocessing.

Given graphs G and H, Subgraph Test asks if G has a subgraph isomorphic to the pattern H. Observe that, for every fixed pattern graph H, Subgraph Test is polynomial-time solvable, as we can test each of the $|V(G)|^{|V(H)|}$ mappings from the vertices of H to the vertices of G, resulting in a polynomial-time algorithm. Therefore, studying the restrictions of Subgraph Test to fixed H does not allow us to make a distinction between easy and hard patterns. We can get a more useful framework if we restrict Subgraph Test to a fixed *class* of patterns. For every graph class $\mathcal{F}$, let $\mathcal{F}$-Subgraph Test be the special case of the problem where H is restricted to be in $\mathcal{F}$. For example, if $\mathcal{F}$ is the set of all matchings (1-regular graphs), then $\mathcal{F}$-Subgraph Test is the polynomial-time solvable maximum matching problem; if $\mathcal{F}$ is the set of all cliques, then $\mathcal{F}$-Subgraph Test is the NP-hard Clique problem. Our goal is to understand which classes $\mathcal{F}$ make $\mathcal{F}$-Subgraph Test tractable.

We also investigate a well-studied and natural variant of finding subgraphs. Given graphs G and H, and an integer t, Packing asks if G has t vertex-disjoint subgraphs isomorphic to H. Unlike for Subgraph Test, now it makes sense to define the problem H-Packing for a fixed graph H: for example, K_2-Packing is the polynomial-time solvable maximum matching problem and K_3-Packing is the NP-hard vertex-disjoint triangle packing problem. We also define the more general $\mathcal{F}$-Packing problem, where H is restricted to be in $\mathcal{F}$.

*This work was partially supported by the European Research Council through starting grant 306992 "Parameterized Approximation" and grant 280152 "PARAMTIGHT: Parameterized complexity and the search for tight complexity results" and OTKA grant NK105645.

†University of Bergen, Norway, bart.jansen@ii.uib.no

‡Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary, dmarx@cs.bme.hu

Kernels and Turing kernels. Besides looking at the polynomial-time solvability of these problems, we also explore the possibility of efficient preprocessing algorithms, as defined by the notion of polynomial kernelization in parameterized complexity [19, 22, 42]. We can naturally associate a parameter k to each instance measuring the size of the solution we are looking for, that is, we define the parameter $k := |V(H)|$ for SUBGRAPH TEST and $k := t \cdot |V(H)|$ for PACKING. We say that a problem with parameter k is *fixed-parameter tractable (FPT)* if it is solvable in time $f(k) \cdot n^{O(1)}$ for some computable function f. The fixed-parameter tractability of various cases of SUBGRAPH TEST is a classical topic of the parameterized complexity literature. It is known that $\mathcal{F}$-SUBGRAPH TEST is FPT if $\mathcal{F}$ is the set of paths [2, 6, 37, 51] and, more generally, if $\mathcal{F}$ is a set of graphs of bounded treewidth [2, 24]. The case where $\mathcal{F}$ is the set of all bicliques (complete bipartite graphs), corresponding to the BICLIQUE problem, was a tantalizing open problem for many years. In a recent breakthrough result, Binkai Lin [41] proved that BICLIQUE is W[1]-hard.

In this paper, we study only a specific aspect of fixed-parameter tractability. A *polynomial (many-one) kernelization* is a polynomial-time algorithm that creates an equivalent instance whose size is polynomially bounded by the parameter k. Intuitively, a kernelization is a preprocessing algorithm that does not solve the problem, but assuming that the parameter value is "small" compared to the size of the input, creates a compact equivalent instance by somehow getting rid of irrelevant parts of the input. In the case of SUBGRAPH TEST, we want to create an equivalent instance with size bounded by $|V(H)|^{O(1)}$: if the pattern H is small compared to G, we want to compress the instance to a "hard core" that has size comparable to H. In recent years, the existence of polynomial kernelization for various parameterized problems has become a thoroughly investigated subject. In 2008, Bodlaender et al. [7] built on a theorem by Fortnow and Santhanam [25] to introduce the lower bound technology of OR-compositions, which allows us to show that certain parameterized problems do not admit polynomial kernels, unless NP $\subseteq$ coNP/poly and the polynomial-time hierarchy collapses to the third level [53]. In particular, they showed that LONG PATH (given an undirected graph G and integer k, does G contain a simple path of length k?) does not admit a polynomial kernel under this complexity assumption. This work has been followed by a flurry of results refining this technology [10, 16, 17, 20, 30] and using it to prove negative results for concrete parameterized problems (e.g., [5, 8, 11, 14, 18, 23, 32, 34, 33, 38, 39], see also the recent survey of Lokshtanov et al. [42]). We continue this line of research by trying to characterize which $\mathcal{F}$-SUBGRAPH TEST and $\mathcal{F}$-PACKING problems admit polynomial kernels.

A natural, but less understood variant of kernelization is Turing kernelization. In a Turing kernelization, instead of creating a single compact instance in polynomial time, we want to solve the instance in polynomial time having access to an oracle solving instances of size $k^{O(1)}$ in constant time. This form of kernelization can be also thought of as some kind of preprocessing: we want to spend polynomial time to preprocess the instance in such a way that the time-consuming part of the work needs to be done on compact instances. While Turing kernelization may seem much more powerful than many-one kernels, there are only a handful of examples where Turing kernelization is possible, but many-one kernelization is not [3, 5, 31, 48, 49]. On the other hand, the lower bound technology introduced by Fortnow and Santhanam [25] and Bodlaender et al. [7] *does not* say anything about the possibility of Turing kernels and therefore we know very little about the limits of Turing kernelization. In fact, even the basic question whether LONG PATH admits a Turing kernel is open (cf. [31]). Hermelin et al. [29] tried to deal with this situation by developing a completeness theory based on certain fundamental satisfiability problems that can be shown to be fixed-parameter tractable by simple branching argument, but for which the existence of polynomial (Turing) kernels is unlikely. They introduced the notion of WK[1]-hardness, which can be interpreted as evidence that the problem is unlikely to admit a polynomial Turing kernel.[1] Unfortunately, Hermelin et al. [29] were unable to prove any hardness result for LONG PATH; its WK[1]-hardness remains an open question. In this paper, we are working under the assumption that LONG PATH admits no polynomial Turing kernel and interpret the existence of a polynomial-parameter transformation from LONG PATH to our problem as evidence for the nonexistence of polynomial Turing kernels. Problems for which such a transformation exists will be called LONG PATH-*hard*.

Our results. In this paper, we restrict our study of $\mathcal{F}$-PACKING and $\mathcal{F}$-SUBGRAPH TEST to hereditary classes $\mathcal{F}$, that is, to classes that are closed under taking induced subgraphs.

The polynomial-time solvability of H-PACKING is well understood: if every component of H has at most two vertices, then it is a matching problem (hence polynomial-time solvable) and Kirkpatrick and Hell [36] proved that H-PACKING is NP-hard for every other H.

[1] It is known [29, Lemma 2] that the existence of a polynomial-size many-one kernel for a WK[1]-hard problem implies NP $\subseteq$ coNP/poly.

It follows that $\mathcal{F}$-Packing is polynomial-time solvable if every component of every graph in $\mathcal{F}$ has at most two vertices, and is NP-hard otherwise. For every fixed H, we can formulate H-Packing as a special case of finding t disjoint sets of size $|V(H)|$ each. Hence the problem admits a polynomial kernel of size $t^{O(|V(H)|)}$ using, for example, standard sunflower kernelization arguments [16, Appendix A]. However, the exponent of the bound on the kernel size depends on the size of H. Therefore, it does not follow that $\mathcal{F}$-Packing admits a polynomial kernel for every fixed class $\mathcal{F}$, as $\mathcal{F}$ may contain arbitrarily large graphs.

Our first result characterizes those hereditary classes $\mathcal{F}$ for which $\mathcal{F}$-Packing admits a polynomial kernel. Interestingly, it seems that Turing kernels are not more powerful for this family of problems: we get the same positive and negative cases with respect to both notions. Let us call a connected bipartite graph b-*thin* if the smaller partite class has size at most b. We say that a graph H is a-*small/*b-*thin* if every component of H either has at most a vertices, or is a b-thin bipartite graph (we emphasize that it is possible that H has components of both types). A graph class $\mathcal{F}$ is small/thin if there are $a, b \geq 0$ such that every graph in $\mathcal{F}$ is a-small/b-thin.

Main Theorem A. *Let $\mathcal{F}$ be a hereditary class of graphs. If $\mathcal{F}$ is small/thin, then $\mathcal{F}$-*Packing *admits a polynomial (many-one) kernel. If $\mathcal{F}$ does not have this property, then $\mathcal{F}$-*Packing *admits no polynomial kernel, unless NP $\subseteq$ coNP/poly, and moreover it is also* WK[1]*-hard,* W[1]*-hard, or* Long Path*-hard.*

Theorem A gives a complete characterization of the hereditary families for which $\mathcal{F}$-Packing admits a polynomial kernel. It is well known that many problems related to packing small graphs/objects admit polynomial kernels (most of the research is therefore on understanding the exact degree of the polynomial bound [1, 12, 16, 30, 46]), but we are not aware of any previous result showing that thin bipartite graphs have similar good properties. This revelation about thin bipartite graphs highlights the importance of looking for dichotomy theorems such as Theorem A: while proving a complete characterization of the positive and negative cases, we *necessarily* have to uncover all the important algorithmic ideas relevant to the family of problems we study. Indeed, our goal was not to prove a result specific to the kernelization of thin bipartite graphs, but it turned out that one cannot avoid proving this result in a complete characterization. The negative part of Theorem A shows that these two algorithmic ingredients (handling small components and thin bipartite graphs) cover all the relevant algorithmic ideas and *any* hereditary class $\mathcal{F}$ that cannot be handled by these ideas leads to a hard problem.

For $\mathcal{F}$-Subgraph Test, we first prove a dichotomy theorem characterizing the randomized polynomial-time solvable and NP-hard cases. We say that $\mathcal{F}$ is *matching-splittable* if there is a constant c such that every $H \in \mathcal{F}$ has a set S of at most c vertices such that every component of $H - S$ has at most 2 vertices.

Main Theorem B. *Let $\mathcal{F}$ be a hereditary class of graphs. If $\mathcal{F}$ is matching-splittable, then $\mathcal{F}$-*Subgraph Test *can be solved in randomized polynomial time. If $\mathcal{F}$ does not have this property, then $\mathcal{F}$-*Subgraph Test *is NP-hard.*

The reason why randomization appears in Theorem B is the following. Given graphs G and $H \in \mathcal{F}$, first we try every possible location where the set $S \subseteq V(H)$ can appear in $V(G)$ in a solution; as $|S| \leq c$, there are $|V(G)|^c$ possibilities to try. Having fixed the location of S, we need to locate every component of $H - S$. As each such component is an edge or a single vertex, this looks like a matching problem, but here we have an additional restriction on how the endpoints of the edges should be attached to S. We can encode these neighborhood conditions using a bounded number of colors and get essentially a colored matching problem, which can be solved in randomized polynomial time using the algorithm of Mulmuley, Vazirani, and Vazirani [47] for finding perfect matchings of exactly a certain weight. The negative side of Theorem B can be obtained by observing (using an application of Ramsey arguments) that if $\mathcal{F}$ is not matching-splittable, then $\mathcal{F}$ contains all cliques, all bicliques, all disjoint unions of triangles, or all disjoint unions of length-two paths; in each case, the problem is NP-hard. The authors are somewhat puzzled that the clean characterization of Theorem B has apparently not been observed so far in the literature: it is about the classical question of polynomial-time solvability of finding subgraphs and the proof uses techniques that are decades old. We may attribute this to the fact that while dichotomy theorems for fixed classes $\mathcal{F}$ of graphs exist (e.g., [13, 15, 27, 28, 35, 40, 52]), perhaps it is not yet widely realized that such results are possible and aiming for them is a doable goal. We hope our paper contributes to the more widespread recognition of the feasibility of this line of research.

In Theorem A, we have observed that Turing kernels are not more powerful than many-one kernels for $\mathcal{F}$-Packing. The situation is different for $\mathcal{F}$-Subgraph Test: there are classes $\mathcal{F}$ for which $\mathcal{F}$-Subgraph Test admits a polynomial Turing kernel, but has no polynomial many-one kernel, unless NP $\subseteq$ coNP/poly. We characterize the classes $\mathcal{F}$ that admit polynomial

Turing kernels the following way. We say that a graph H is (a, b, c, d)-splittable, if there is a set S of at most c vertices such that every component of $H - S$ either has size at most a or is a b-thin bipartite graph with the additional restriction that the closed neighborhoods of all but d vertices are universal to S (see Section 2.1 for details).

Main Theorem C. *Let $\mathcal{F}$ be a hereditary class of graphs. If there are $a, b, c, d \geq 0$ such that every $H \in \mathcal{F}$ is (a, b, c, d)-splittable, then $\mathcal{F}$-*Subgraph Test *admits a polynomial Turing kernel. If $\mathcal{F}$ does not have this property, then $\mathcal{F}$-*Subgraph Test *is* WK[1]*-hard,* W[1]*-hard, or* Long Path*-hard.*

In the algorithmic part of Theorem C, the first step is to guess the location of the set $S \subseteq V(H)$ in $V(G)$, giving $|V(G)|^c$ possibilities (this is the reason why in general our Turing kernel is not a many-one kernel). For each guess, locating the components of $H - S$ in G is similar to Theorem A, as we have to handle small components and thin bipartite components, but here we have the additional technicality that we have to ensure that these components are attached to S in a certain way.

For many-one kernels, we do not have a characterization similar to Theorem C. We present some concrete positive and negative results showing that a complete characterization of $\mathcal{F}$-Subgraph Test with respect to many-one kernels would be much more delicate than Theorem C. The simple algorithmic idea used in Theorem C, guessing the location of S, fails for many-one kernels and it seems that we have to make extreme efforts (whenever it is possible at all) to replace this step with adhoc arguments.

Our techniques. The proofs of Theorems A–C all follow the same pattern. First, we define a certain graph-theoretic property and devise an algorithm for the case when $\mathcal{F}$ has this property. As described above, the algorithmic part of Theorem B is based on the randomized matching algorithm of Mulmuley, Vazirani, and Vazirani [47]. For Theorems A and C, the algorithm is a marking procedure: for each component, we mark a bounded number of vertices such that we can always find a copy of this component using only these vertices even if the other components already occupy an unknown but small set of vertices. Therefore, if there is a solution, then there is a solution using only this set of marked vertices. The kernel is obtained by restricting the graph to this set of vertices. For small components, we use the Sunflower Lemma of Erdős and Rado [21] (similarly as it is used in the kernelization of other packing problems, cf. [16]). For thin bipartite graphs, the marking procedure is a branching algorithm specifically designed for this class of graphs. At some point in the algorithm, we crucially use that the component is b-thin: we find a biclique with b vertices on one side and many vertices on the other side, and then we argue that the component is a subgraph of this biclique.

For the hardness results of Theorems A–C, first we prove that if $\mathcal{F}$ does not have the stated property, then $\mathcal{F}$ contains every graph from one of the basic families of hard graphs. These hard families include cliques, bicliques, paths, odd cycles with a high-degree vertex, and subdivided stars (see Section 3.2). To prove that a hard family appears in $\mathcal{F}$, we use Ramsey results (including a recent path vs. induced path vs. biclique result of Atminas, Lozin, and Razgon [4]) and a graph-theoretic analysis of what, for example, a large nonbipartite graph without large cliques and long induced paths can look like. For each hard family, we then claim a lower bound on the problem. Most of these lower bounds take the form of a relatively standard polynomial-parameter transformation from Set Cover parameterized by the size of the universe; here the value of our contribution is not in the details of the reduction, but in realizing that these are the hard families of graphs whose hardness exhaustively explain the hard cases of the problem.

The basic technique to obtain negative evidence for the existence of many-one kernels is the method of *OR-cross-composition* [10], which refines the original OR-composition framework [7]. The negative results that we present for the existence of many-one kernels for $\mathcal{F}$-Subgraph Test use a specific form of this technique that we name *OR-cross-composition by reduction with a canonical template.* The idea is to start from an NP-hard graph problem L for which a family of polynomial-size *canonical template graphs* exists, such that for every n, the instances of length n are induced subgraphs of the n-th graph in this family. This allows length-n inputs $x_1, \ldots, x_t$ to be merged into one through their common canonical supergraph of size poly(n), as opposed to the trivial $t \cdot n$, which facilitates an OR-cross-composition. Canonical template graphs were first used for this purpose by Bodlaender et al. [9, Theorem 11].

2 Outline

In this section we present a more detailed overview of the results of the paper. We also describe the main technical parts of the proofs. The proofs of Theorems A–C all follow the same pattern:

(1) We define the property separating the positive and negative cases.
(2) We prove an algorithmic result for the positive cases.

(3) We prove a purely combinatorial result stating that if a class $\mathcal{F}$ does not satisfy the property, then $\mathcal{F}$ is a superset of one of the classes appearing on a short list of basic hard classes.
(4) We prove a hardness result for each basic hard class on the list.

The structure of this section follows these steps: for each step, we go through the relevant definitions and state the results proved later in the paper.

2.1 Characterizing properties We say that a graph is *c-matching-splittable* if there is a set $S \subseteq V(H)$ of at most c vertices such that every component of $H-S$ has at most two vertices. We say that a class $\mathcal{F}$ of graphs is c-matching-splittable if every $H \in \mathcal{F}$ has this property, and we say that $\mathcal{F}$ is matching-splittable if $\mathcal{F}$ is c-matching-splittable for some $c \geq 0$. In Theorem B, this is the condition for randomized polynomial-time solvability. Clearly, a matching is 0-matching-splittable and a matching plus a universal vertex is 1-matching-splittable. On the other hand, the class containing the disjoint unions of arbitrarily many triangles is not c-matching-splittable for any $c \geq 0$, as S would need to contain at least one vertex from each triangle.

In Theorem A, the condition that we need is that every component is either small or a thin bipartite graph. We say that a graph H is *a-small/b-thin* if every component of H has at most a vertices or is a b-thin bipartite graph (that is, a bipartite graph with one of the partite classes having size at most b). Note that H can have both types of components. For example, if H is the disjoint union of an arbitrary number of triangles and stars of arbitrary size, then it is 3-small/1-thin. We say that class $\mathcal{F}$ is a-small/b-thin if every graph $H \in \mathcal{F}$ has this property and say that $\mathcal{F}$ is small/thin if it is a-small/b-thin for some $a, b \geq 0$. The characterization property that we need for Theorem C is a somewhat technical generalization of being a-small/b-thin.

Definition 3. *We say that a graph H is (a,b,c,d)-*splittable *if it has a vertex set $S \subseteq V(H)$ of size at most c such that:*

(1) each connected component of $H-S$ on more than a vertices is bipartite and has a partite class of size at most b, and
(2) in each connected component C of $H-S$, the number of vertices whose closed neighborhood in $G[C]$ is not *universal to $N_H(C) \cap S$ is at most d.*

We say that such a set $S \subseteq V(H)$ realizes *the (a,b,c,d)-split of H. Family $\mathcal{F}$ is (a,b,c,d)-splittable if every $H \in \mathcal{F}$ is a (a,b,c,d)-splittable. Family $\mathcal{F}$ is splittable if there are constants a,b,c,d such that $\mathcal{F}$ is (a,b,c,d)-splittable.*

Observe that being a-small/b-thin is exactly the same as being $(a,b,0,0)$-splittable and being c-matching-splittable is exactly the same as being $(2,0,c,2)$-splittable. We prefer to use the terms a-small/b-thin and c-matching-splittable for these special cases, as they are more descriptive.

Given an a-small/b-thin graph H, adding a set S of c universal vertices results in an $(a,b,c,0)$-splittable graph H'. If C is a component of H having at most a vertices and we remove from H' any set of edges between C and S, then the resulting graph is (a,b,c,a)-splittable. The closed neighborhoods of the a vertices in C may no longer be universal to $N_H(C) \cap S$ after the edge removals, which is compensated by the fourth entry in the tuple. Let now C be a b-thin bipartite component of H, let A be the smaller side and let B be the larger side of C. Observe that Definition 3 not only requires that all but d vertices of C are universal to $N_H(C) \cap S$, but even the closed neighborhoods in $G[C]$ have to be universal. Therefore, removing even a single edge between a vertex v of C and S can ruin the property, as it "contaminates" all the neighbors of v. If we remove a single edge between some $x \in B$ and S, then the graph is still $(a,b,c,b+1)$-splittable: there are at most $b+1$ vertices in C whose neighborhood is not universal to S, namely x and some of the vertices of A. On the other hand, if we remove a single edge between some $y \in A$ and S, then the graph may not be (a,b,c,d)-splittable for arbitrary large d: if y has degree d, then y and all its neighbors have the property that their closed neighborhoods are not universal to S.

Note that the definition *does not* require that the closed neighborhood of (all but d of) the vertices are universal to S, it requires universality only to $N_H(C) \cap S$. Suppose that H_1 and H_2 are two graphs with S_i realizing an (a,b,c,d)-split of H_i for $i = 1,2$. The disjoint union of H_1 and H_2 is $(a,b,2c,d)$-splittable, as realized by $S_1 \cup S_2$: the vertices in a b-thin component of H_1 need to be universal only to (a certain part of) S_1, as C has no edge to S_2.

3.1 Algorithms In the algorithmic part of Theorem B, we need to solve Subgraph Test in the case that H is c-matching-splittable for some set S of at most c vertices. As described in the introduction, we guess the location of S and then solve the resulting constrained matching problem. The main technical engine in the algorithm is the classic algebraic matching algorithm due to Mulmuley, Vazirani, and Vazirani [47]. It can be used to obtain randomized algorithms for various colored versions of matching (see, for example, [43, 44]). We need the following variant.

Theorem 3.1. *Given a multigraph G with a (not nec-*

essary proper) coloring of the edges with a set C of colors and function $f : C \to \mathbb{Z}^+$, there is a randomized algorithm with false negatives that decides in time $(|V(G)| + |E(G)|)^{O(|C|)}$ if G has a matching containing exactly $f(i)$ edges of color i for every $i \in C$.

By a randomized algorithm with false negatives, we mean an algorithm that is always correct on NO-instances, but which may incorrectly reject a YES-instance with probability at most $\frac{1}{2}$. Equipped with Theorem 3.1, we can prove the algorithmic part of Theorem B.

THEOREM 3.2. $\mathcal{F}$-SUBGRAPH TEST *is (randomized) polynomial-time solvable if $\mathcal{F}$ is matching-splittable.*

The polynomial kernel in the positive part of Theorem A is obtained by a marking procedure that finds a polynomially bounded subset of vertices in G that surely contains a solution, if a solution exists at all. Let us first explain briefly how the standard technique of sunflowers can be used for this marking procedure if every component of H has at most a vertices. We need the Sunflower Lemma of Erdős and Rado [21]. A collection $\mathcal{S}$ of sets is called a *sunflower* if the pairwise intersection $S_1 \cap S_2$ is the same set C for any two distinct $S_1, S_2 \in \mathcal{S}$. Then this intersection C is the *core* of the sunflower; the sets $S \setminus C$ for $S \in \mathcal{S}$ are the *petals* of the sunflower.

LEMMA 3.1. ([21], CF. [22, LEMMA 9.7]) *Let k and m be nonnegative integers and let $\mathcal{S}$ be a system of sets of size at most m over a universe U. If $|\mathcal{S}| \geq m!(k-1)^m$, then there is a sunflower in $\mathcal{S}$ with k petals. Furthermore, for every fixed m there is an algorithm that computes such a sunflower in time polynomial in $(k + |\mathcal{S}|)$.*

Let H, G, and $t \geq 1$ form an instance of PACKING; the solution we are looking for has $k := t \cdot |V(H)|$ vertices. Let C be a component of H having size at most a. First, we enumerate every subset of $|V(C)| \leq a$ vertices in G where C appears; the length of this list is polynomial in the size of G if a is a fixed constant. We would like to reduce the length of this list: we would like to have a shorter list of candidate locations where C can appear in a solution, such that the length of the list is polynomially bounded in k. We argue the following way. As long as the length of the list is at least $a!(k+1)^a$, we can find a sunflower with $k+2$ petals among the sets in the list. We claim that we can choose any set S from this sunflower and throw it out of the list. Suppose that there is a solution where the component C is mapped exactly to this set $S \subseteq V(G)$. As the solution uses only k vertices of G and the petals of the sunflower are disjoint, there is another set S' among the remaining $k + 1$ sets of the sunflower whose petal is disjoint from the solution. Therefore, we can modify the solution such that C is mapped to S' instead of S, which means that the set S cannot be essential to the solution and can be safely removed from the list of candidate locations for C. Repeating this argument, we eventually get a list of at most $a!(k + 1)^a$ candidate locations for each component of H, thus we can reduce the problem to an induced subgraph of G whose size, for a fixed constant a, is polynomial in k.

If H has b-thin components, then the Sunflower Lemma cannot be applied, as the size of such a component can be arbitrarily large (and it is the size of the component that appears in the exponent in the argument above). Therefore, we develop a marking procedure specifically designed for thin bipartite graphs. As an illustration, we present here the main idea on the special case of packing thin bicliques, that is, on graphs $K_{b,\ell}$ for some fixed $b \geq 1$. The crucial ingredient for the kernel for biclique packing is the following lemma.

LEMMA 3.2. *For every fixed b there is a polynomial-time algorithm that, given a graph G and integers $\ell > b$ and $k \geq \ell + b$, computes a set X of size $\mathcal{O}(k^{4b})$ such that for every $Z \subseteq V(G)$ of size at most k, if $G - Z$ contains a $K_{b,\ell}$ subgraph, then $G[X] - Z$ contains a $K_{b,\ell}$ subgraph.*

Before proving the lemma, we show how it leads to a polynomial kernel for biclique packing. To reduce the size of an instance that asks whether G contains t disjoint $K_{b,\ell}$ subgraphs for $\ell > b$, we define $k := t \cdot (b + \ell)$ and invoke the lemma to compute a set X of size $\mathcal{O}(k^{4b})$. We then output $G[X]$ as the kernelized instance. If G contains a packing of t disjoint $K_{b,\ell}$ subgraphs, then while the packing contains a biclique C using a vertex in $V(G) \setminus X$, we let Z be the $(t-1)(b+\ell)$ other vertices in the packing, apply the guarantee of the lemma to find a biclique model C' in $G[X]$ avoiding Z, and replace C in the packing by C'. Iterating the argument results in a packing of bicliques in $G[X]$, proving that the reduced instance is equivalent to the original one.

To facilitate a recursive algorithm, we actually prove a generalization of Lemma 3.2. To state the generalization we need the following terminology. For disjoint sets $A', B' \subseteq V(G)$ and $\ell > b$ we say that a $K_{b,\ell}$ subgraph in G extends (A', B') if the side-b partite class is a superset of A' and the size-ℓ partite class is a superset of B'.

LEMMA 3.3. *For every fixed b there is a polynomial-time algorithm that, given a graph G, integers $\ell > b$ and $k \geq \ell + b$, and disjoint sets $A', B' \subseteq V(G)$ of size at most b, computes a set X of size at most $(3k^2)^{2b-|A' \cup B'|}$ such that for every $Z \subseteq V(G)$ of size at most k, if $G - Z$ contains a $K_{b,\ell}$ subgraph that extends (A', B'), then $G[X] - Z$ contains a $K_{b,\ell}$ subgraph.*

Proof. The main idea behind the algorithm is to make progress in recursive calls by increasing the size of $A' \cup B'$, thereby restricting the type of bicliques that have to be preserved in the set X. Throughout the proof we use the fact that if $Z \subseteq V(G)$ and there is a $K_{b,\ell}$-subgraph in $G - Z$ that extends (A', B'), then $Z \cap (A' \cup B') = \emptyset$, the size-$b$ partite class consists of common neighbors of B', while the size-ℓ partite class consists of common neighbors of A'. Let us point out that the lemma requires that $G[X] - Z$ contains an $K_{b,\ell}$-subgraph, but it does not require it to extend (A', B').

Case 1. If $|A'| = b$, then we choose X as $A' \cup B'$ together with $k + \ell$ common neighbors of A' (or less, if there are fewer), for a total size of at most $2b+(k+\ell) \leq 3k$. Let $Z \subseteq V(G)$ have size at most k. If there is a $K_{b,\ell}$-subgraph H in $G-Z$ that extends (A', B'), then all vertices in the size-ℓ partite class are common neighbors of A'. If all vertices of H are contained in $G[X]$, then the biclique subgraph H also exists in $G[X]-Z$. If not, then the set A' had at least $k + \ell$ common neighbors (otherwise they were all preserved in X). Since Z contains at most k of them, any ℓ of the remaining vertices in X combines with A' to form a $K_{b,\ell}$-subgraph in $G[X] - Z$.

Case 2.a. If $|B'| = b$, $|A'| < b$, and the set B' has at least $k + \ell$ common neighbors, then we choose X containing $k + \ell$ of these common neighbors together with B' itself. For any $Z \subseteq V(G)$ of size at most k, if a biclique extending (A', B') exists in $G - Z$ then Z avoids at least ℓ common neighbors of B' in X. Together with B', these form a $K_{b,\ell}$ subgraph in $G[X] - Z$. Note that this $K_{b,\ell}$ does not extend (A', B'), but this is not required by the lemma.

Case 2.b. If $|B'| = b$, $|A'| < b$, and the set B' has less than $k + \ell \leq 2k$ common neighbors $T := \bigcap_{v \in B'} N_G(v)$, then a $K_{b,\ell}$-subgraph extending (A', B') has its size-b side within T. For each $a \in T \setminus (A' \cup B')$, add a to A' and recurse. Let X be the union of the recursively computed sets. If there is a biclique in $G - Z$ extending (A', B'), then there is an $a \in T \setminus (A' \cup B')$ such that it extends $(A' \cup \{a\}, B')$, and the correctness guarantee for that recursive call yields a biclique in $G[X] - Z$. The measure $2b - |A' \cup B'|$ drops in each recursive call and we recurse on at most $2k$ instances, giving a bound of $2k \cdot (3k^2)^{2b-|A' \cup B'|-1} \leq (3k^2)^{2b-|A' \cup B'|}$ on $|X|$.

Case 3. In the remaining cases we have $|A'|, |B'| < b$. We greedily compute a maximal set of $K_{b,\ell}$ subgraphs that extend (A', B') and pairwise intersect only in $A' \cup B'$. Since b is constant, this can be done in polynomial time by guessing all possible locations for the remaining vertices in the size-b partite class and testing whether the resulting vertices are adjacent to B' and have sufficient common neighbors to realize the other partite class. Two things can happen.

Case 3.a. If we find $k + 1$ distinct $K_{b,\ell}$ subgraphs that pairwise intersect only in (A', B'), then we output X containing the union of these subgraphs, which has size at most $(k+1)(\ell+b) \leq 2k^2$. If a $K_{b,\ell}$-subgraph extending (A', B') exists in $G - Z$ for some $Z \subseteq V(G)$ of size k, then Z intersects at most k of the extensions. Hence one extension avoids Z and combines with A', B' to form a $K_{b,\ell}$-subgraph in $G[X] - Z$.

Case 3.b. If there are at most k of such extensions, then let T contain the at most $k(\ell + b) \leq k^2$ vertices in their union. By the maximality of the packing, any extension of (A', B') uses a vertex in $T \setminus (A' \cup B')$. For each $v \in T \setminus (A' \cup B')$, recurse twice: once for adding v to A' and once for adding v to B'. We let X be the union of the recursively computed sets. If there is a $K_{b,\ell}$ subgraph in $G - Z$ for some $Z \subseteq V(G)$ of size at most k, then it extends $(A' \cup \{v\}, B')$ or $(A', B' \cup \{v\})$ for some $v \in T \setminus (A' \cup B')$. The correctness guarantee for that branch of the recursion guarantees the existence of $K_{b,\ell}$ in $G[X] - Z$. As the measure $2b - |A' \cup B'|$ drops in each recursive call, while we branch in at most $2|T| \leq 2k(\ell + b) \leq 2k^2$ directions, the size of X is bounded by $2k^2 \cdot (3k^2)^{2b-|A' \cup B'|-1} \leq (3k^2)^{2b-|A' \cup B'|}$.

The generalization from b-thin bicliques to general b-thin bipartite graphs makes the scheme described above much more technical. Let us point out that the large side of a b-thin bipartite graph can be partitioned into at most 2^b classes according to its neighborhood in the small side. Therefore, intuitively, a b-thin bipartite graph can be seen as 2^b different b-thin bicliques joined together, which makes it plausible that such a generalization exists.

THEOREM 3.3. *If $\mathcal{F}$ is a hereditary class of graphs that is small/thin, then $\mathcal{F}$-*PACKING *admits a polynomial many-one kernel.*

For the algorithmic part of Theorem C, we have to guess the location of the set S realizing the (a, b, c, d)-split and then take into account the universality restrictions. This introduces another layer of technical difficulties, but no new conceptual ideas are needed. Moreover, because of this guessing step, the kernel is no longer many-one, but it is a Turing kernel.

THEOREM 3.4. *If $\mathcal{F}$ is a hereditary class of graphs that is splittable, then $\mathcal{F}$-*SUBGRAPH TEST *admits a polynomial Turing kernel.*

3.2 Hard families We define several specific classes of graphs and show hardness results for these classes.

Then we show that if a class does not have the property of, say, being splittable, then it is a superset of at least one hard class, hence hardness follows for every class that does not have this property.

First, we define the following graphs (see Figure 1).

- $\mathsf{Path}(\ell)$ is the path of length ℓ, which consists of ℓ edges and $\ell+1$ vertices. It is sometimes denoted $P_{\ell+1}$ for brevity.
- $\mathsf{Clique}(n)$ is the clique on n vertices (while describing hard families, we use $\mathsf{Clique}(n)$ instead of the more standard K_n for consistency of notation).
- $\mathsf{Biclique}(n)$ is the balanced biclique $K_{n,n}$ on $n+n$ vertices.
- $\mathsf{2\text{-}broom}(s,n)$ is obtained from a length-s path by adding n pendant vertices to each of the two endpoints of the path.
- $\mathsf{OperaHouse}(s,n)$ is obtained from a length-s path by adding n vertices that are adjacent to both endpoints of the path.
- $\mathsf{Fountain}(s,n)$ is obtained from a length-s cycle by adding n pendant vertices to one vertex on the cycle.
- $\mathsf{LongFountain}(s,t,n)$ is obtained from a length-s cycle by adding a path of length t, identifying one endpoint with a vertex on the cycle and adding n pendant vertices to the other endpoint.
- $\mathsf{SubDivStar}(n)$ is obtained from a star with n leaves by subdividing each edge once.
- $\mathsf{SubDivTree}(s,n)$ is obtained from a star with n leaves by subdividing each edge $s-1$ times and attaching n pendant vertices to each leaf.
- $\mathsf{DiamondFan}(n)$ is obtained from n copies of $K_{2,n}$ by taking one degree-n vertex from each copy and identifying them into a single vertex.

We can define families of these graphs the obvious way:

$$\begin{aligned}
&\mathcal{F}_{\mathsf{Path}} = \{\mathsf{Path}(i) \mid i \geq 1\}\\
&\mathcal{F}_{\mathsf{Clique}} = \{\mathsf{Clique}(i) \mid i \geq 1\}\\
&\mathcal{F}_{\mathsf{Biclique}} = \{\mathsf{Biclique}(i) \mid i \geq 1\}\\
&\mathcal{F}^{s}_{\mathsf{2\text{-}broom}} = \{\mathsf{2\text{-}broom}(s,i) \mid i \geq 1\}\\
&\mathcal{F}^{s}_{\mathsf{Fountain}} = \{\mathsf{Fountain}(s,i) \mid i \geq 1\}\\
&\mathcal{F}^{s,t}_{\mathsf{LongFountain}} = \{\mathsf{LongFountain}(s,t,i) \mid i \geq 1\}\\
&\mathcal{F}^{s}_{\mathsf{OperaHouse}} = \{\mathsf{OperaHouse}(s,i) \mid i \geq 1\}\\
&\mathcal{F}_{\mathsf{SubDivStar}} = \{\mathsf{SubDivStar}(i) \mid i \geq 1\}\\
&\mathcal{F}^{s}_{\mathsf{SubDivTree}} = \{\mathsf{SubDivTree}(s,i) \mid i \geq 1\}\\
&\mathcal{F}_{\mathsf{DiamondFan}} = \{\mathsf{DiamondFan}(i) \mid i \geq 1\}
\end{aligned}$$

To prove that a hard family is contained in every class not satisfying a certain property, we use arguments based on Ramsey theory. The following lemma characterizes hereditary classes that are not matching-splittable. We define $n \cdot H$ to be the graph that contains n disjoint copies of H. (Recall that P_3 is the path on 3 vertices.)

Theorem 3.5. *Let $\mathcal{F}$ be a hereditary graph family that is not matching splittable. Then at least one of the following holds:*

(1) $\mathcal{F}$ is a superset of $\mathcal{F}_{\mathsf{Clique}}$.
(2) $\mathcal{F}$ is a superset of $\mathcal{F}_{\mathsf{Biclique}}$.
(3) $\mathcal{F}$ contains $n \cdot K_3$ for every $n \geq 1$.
(4) $\mathcal{F}$ contains $n \cdot P_3$ for every $n \geq 1$.

Observe that Theorem 3.5 is a tight characterization of matching-splittable graphs: the converse statement is also true, that is, if any of the four statements is true for $\mathcal{F}$, then it is not matching-splittable. Clearly, large cliques and large bicliques are not c-matching-splittable for constant c. Moreover, if every component of a graph has three vertices (that is, it is either a K_3 or P_3), then at least one vertex has to be deleted from each component to decrease the size of every component to at most two vertices, hence $\mathcal{F}$ cannot be c-matching-splittable for constant c in the last two cases either. The following theorem characterizes the hereditary classes that are not small/thin.

Theorem 3.6. *Let $\mathcal{F}$ be a hereditary graph family that is not small/thin. Then $\mathcal{F}$ is a superset of at least one of the following families:*

(1) $\mathcal{F}_{\mathsf{Path}}$,
(2) $\mathcal{F}_{\mathsf{Clique}}$,
(3) $\mathcal{F}_{\mathsf{Biclique}}$,
(4) $\mathcal{F}^{s}_{\mathsf{Fountain}}$ for some odd integer $s \geq 3$,
(5) $\mathcal{F}^{s,t}_{\mathsf{LongFountain}}$ for some odd integer $s \geq 3$ and integer $t \geq 1$,
(6) $\mathcal{F}^{s}_{\mathsf{OperaHouse}}$ for some odd integer $s \geq 1$,
(7) $\mathcal{F}_{\mathsf{SubDivStar}}$, or
(8) $\mathcal{F}^{s}_{\mathsf{2\text{-}broom}}$ for some odd integer $s \geq 1$.

Again, the characterization is tight: we can observe that if $\mathcal{F}$ is a superset of any of these families, then there is no $a, b \geq 0$ such that $\mathcal{F}$ is a-small/b-thin. Note that we cannot leave out any of the eight items from the list: the hereditary closure of, say, $\mathcal{F}^{5,2}_{\mathsf{LongFountain}}$ is not the superset of any of the classes described in the remaining seven items.

Finally, we characterize graphs that are not splittable.

Theorem 3.7. *Let $\mathcal{F}$ be a hereditary graph family that is not splittable. Then at least one of the following holds:*

(1) $\mathcal{F}$ is a superset of $\mathcal{F}_{\mathsf{Path}}$,
(2) $\mathcal{F}$ is a superset of $\mathcal{F}_{\mathsf{Clique}}$,
(3) $\mathcal{F}$ is a superset of $\mathcal{F}_{\mathsf{Biclique}}$,

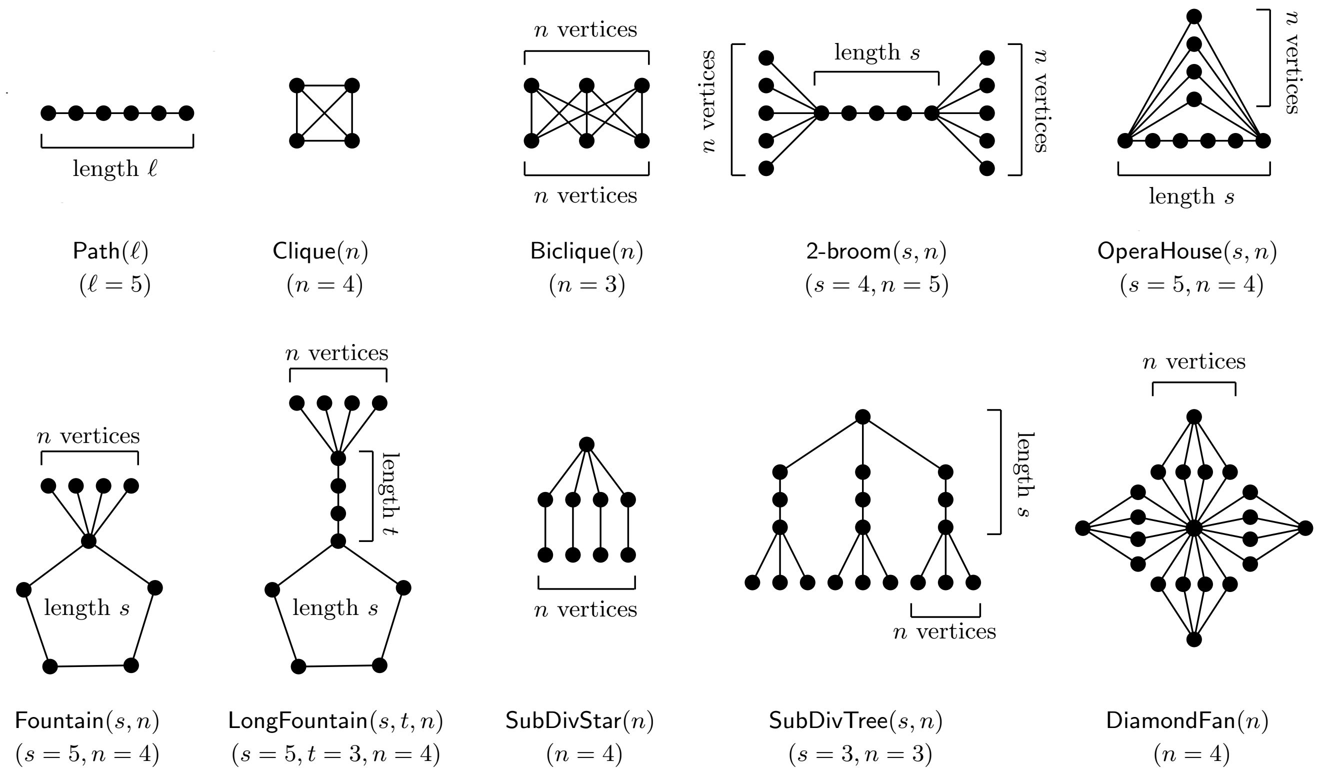

Figure 1: Basic families of graphs.

(4) $\mathcal{F}$ *contains* $n \cdot \mathsf{SubDivStar}(n)$ *for every* $n \geq 1$*,*

(5) there is an odd $s \geq 3$ *such that* $\mathcal{F}$ *contains* $n \cdot \mathsf{Fountain}(s, n)$ *for every* $n \geq 1$*,*

(6) there is an odd $s \geq 1$ *such that* $\mathcal{F}$ *contains* $n \cdot \mathsf{OperaHouse}(s, n)$ *for every* $n \geq 1$*,*

(7) there is an odd $s \geq 1$ *such that* $\mathcal{F}$ *contains* $n \cdot \mathsf{2\text{-}broom}(s, n)$ *for every* $n \geq 1$*,*

(8) there is an odd $s \geq 3$ *and arbitrary* $t \geq 1$ *such that* $\mathcal{F}$ *contains* $n \cdot \mathsf{LongFountain}(s, t, n)$ *for every* $n \geq 1$*,*

(9) $\mathcal{F}$ *is a superset of* $\mathcal{F}^s_{\mathsf{SubDivTree}}$ *for some integer* $s \geq 1$*, or*

(10) $\mathcal{F}$ *is a superset of* $\mathcal{F}_{\mathsf{DiamondFan}}$*.*

We can again verify that the characterization is tight. In particular, let us show that $\mathsf{SubDivTree}(s, n)$ is not (a, b, c, d)-splittable if $n > a + b + c + d$. Suppose that S realizes the (a, b, c, d)-split. As the graph is not b-thin and has more than a vertices, we have that S is not empty. By the pigeonhole principle, there is a vertex v with degree $n+1$ that is not in S, and none of its degree-1 neighbors are in S either. Then v is in a component of size at least $n + 1 > a$ that contains at least $n > d$ vertices that have no neighbors in S. Similarly, suppose that S realizes an (a, b, c, d)-split of $\mathsf{DiamondFan}(n)$ for $n > a + b + c + d$. Again, S is not empty. By the pigeonhole principle, there is a degree-n vertex v that is not in S and has no neighbor in S. The component of this vertex has size more than a and the component has more than d vertices (namely, every neighbor of v) whose closed neighborhood is not universal to S.

3.3 Hardness proofs Let us review the concrete hardness results that we prove, which, by the combinatorial characterizations in Theorems 3.5–3.7, prove the negative parts of Theorems A–C. As mentioned above, Kirkpatrick and Hell [36] fully characterized the polynomial-time solvable cases of H-Packing.

Theorem 3.8. ([36]) H-Packing *is polynomial-time solvable if every connected component of* H *has at most two vertices and NP-complete otherwise.*

It follows from Theorem 3.8 that $\mathcal{F}$-Subgraph Test is NP-hard if $\mathcal{F}$ contains $n \cdot K_3$ for every $n \geq 1$ or if $\mathcal{F}$ contains $n \cdot P_3$ for every $n \geq 1$, as then the problem is more general than K_3-Packing or P_3-Packing, respectively. Also, $\mathcal{F}$-Subgraph Test is NP-hard if $\mathcal{F}$ contains every clique [26, GT7] (it generalizes Clique) or if $\mathcal{F}$ contains every biclique [26, GT24].

For kernelization lower bounds, observe first that if $\mathcal{F}$ contains every clique, then $\mathcal{F}$-Packing and $\mathcal{F}$-

SUBGRAPH TEST are clearly W[1]-hard [19, Theorem 21.2.4] and therefore do not admit a (Turing) kernel of any size, unless FPT = W[1] and the Exponential Time Hypothesis fails [19, Chapter 29]. A recent result of Lin [41] shows that if $\mathcal{F}$ contains every biclique, then $\mathcal{F}$-PACKING and $\mathcal{F}$-SUBGRAPH TEST are also W[1]-hard. Since the parameterized CLIQUE and BICLIQUE problems are NP-hard and OR-compositional [7], it follows from standard kernelization lower bound machinery that if $\mathcal{F}$-PACKING or $\mathcal{F}$-SUBGRAPH TEST has a polynomial (many-one) kernel when $\mathcal{F}$ contains every clique or biclique, then NP $\subseteq$ coNP/poly. To complete the proof of the negative parts of Theorems A and C, we prove the following two sets of WK[1]-hardness results.

THEOREM 3.9. *The* $\mathcal{F}$-PACKING *problem is* WK[1]-*hard under polynomial-parameter transformations if* $\mathcal{F}$ *is a superset of any of the following families:*

(1) $\mathcal{F}_{\mathsf{SubDivStar}}$,
(2) $\mathcal{F}^{s,t}_{\mathsf{LongFountain}}$ *for some integer* $t \geq 1$ *and some* odd *integer* $s \geq 3$,
(3) $\mathcal{F}^{s}_{\mathsf{2\text{-}broom}}$ *for some odd integer* $s \geq 1$,
(4) $\mathcal{F}^{s}_{\mathsf{Fountain}}$ *for some odd integer* $s \geq 3$, *or*
(5) $\mathcal{F}^{s}_{\mathsf{OperaHouse}}$ *for some odd integer* $s \geq 1$.

THEOREM 3.10. *The* $\mathcal{F}$-SUBGRAPH TEST *problem is* WK[1]-*hard under polynomial-parameter transformations if* $\mathcal{F}$ *is a superset of any of the following families:*

(1) $\mathcal{F}_{\mathsf{DiamondFan}}$, *or*
(2) $\mathcal{F}^{s}_{\mathsf{SubDivTree}}$ *for some integer* s.

All WK[1]-hardness proofs are by reduction from UNIFORM EXACT SET COVER (n), where the parameter equals the size of the universe on which the set system is defined. The uniform variant, in which all sets have the same size, is particularly useful for proving these results. We prove the WK[1]-hardness of the problem by a two-stage transformation from EXACT SET COVER (n) [29], first introducing a small number of new elements to ensure that solutions exist that contain a prescribed number of sets, and then using this knowledge to introduce another small number of elements that can be added to the sets to make the system uniform.

3.4 Many-one kernels We do not have a complete characterization of the existence of many-one kernels for $\mathcal{F}$-SUBGRAPH TEST. The authors believe that if such a characterization is possible, then it has to be significantly more delicate than the characterization of Turing kernels in Theorem C and both the positive and the negative parts should involve a larger number of specific cases. We present two lower bounds and two upper bounds to show the difficulties that arise (see also Figure 2). The following two theorems give the lower bounds.

THEOREM 3.11. *Let* $\mathcal{F}$ *be any hereditary graph family containing all graphs of the form* $H' + \ell \cdot K_3$, *where* $\ell \geq 1$ *and* $H' \in \mathcal{F}_{\mathsf{SubDivStar}}$. *Then* $\mathcal{F}$-SUBGRAPH TEST *does not admit a polynomial many-one kernel unless NP* $\subseteq$ *coNP/poly.*

THEOREM 3.12. *Let* $\mathcal{F}$ *be any hereditary graph family containing all graphs of the form* $H' + H'' + \ell \cdot P_3$, *where* $\ell \geq 1$ *and* $H', H'' \in \mathcal{F}_{\mathsf{SubDivStar}}$. *Then* $\mathcal{F}$-SUBGRAPH TEST *does not admit a polynomial many-one kernel unless NP* $\subseteq$ *coNP/poly.*

Observe that the graph families described by these theorems are $(3, 0, 2, 2)$-splittable: letting S contain the (at most two) centers of the subdivided stars, the connected components that remain after removing S have at most three vertices. Every leg of a subdivided star becomes a component of size two in which one of the vertices is universal to S and the other is not; hence the closed neighborhoods of the two vertices are not universal to S. The $\mathcal{F}$-SUBGRAPH TEST problem for these families therefore has polynomial Turing kernels by Theorem 3.4, highlighting the difference between many-one and Turing kernelization for $\mathcal{F}$-SUBGRAPH TEST. The following two theorems give upper bounds.

THEOREM 3.13. *Let* $\mathcal{F}$ *be the hereditary closure of the family containing all graphs of the form* $H' + \ell \cdot K_3$, *where* $\ell \geq 1$ *and* $H' \in \mathcal{F}^{3}_{\mathsf{Fountain}}$. *Then* $\mathcal{F}$-SUBGRAPH TEST *admits a polynomial many-one kernel.*

THEOREM 3.14. *Let* $\mathcal{F}$ *be the hereditary closure of the family containing all graphs of the form* $H' + \ell \cdot P_3$, *where* $\ell \geq 1$ *and* $H' \in \mathcal{F}_{\mathsf{SubDivStar}}$. *Then* $\mathcal{F}$-SUBGRAPH TEST *admits a polynomial many-one kernel.*

Comparing Theorem 3.11 to Theorem 3.13, we find that changing the type of the single large component from a subdivided star to a fountain crosses the threshold for the existence of a polynomial kernel, even though both types of graphs can be reduced to constant-size components by a single vertex deletion. Comparing Theorem 3.12 to Theorem 3.14 we see that decreasing the number of subdivided star components from two to one makes a polynomial kernel possible. While the definition of splittable graph families that characterizes the existence of polynomial Turing kernels for $\mathcal{F}$-SUBGRAPH TEST is robust under increases by constants, this is clearly not the case for the many-one complexity of $\mathcal{F}$-SUBGRAPH TEST.

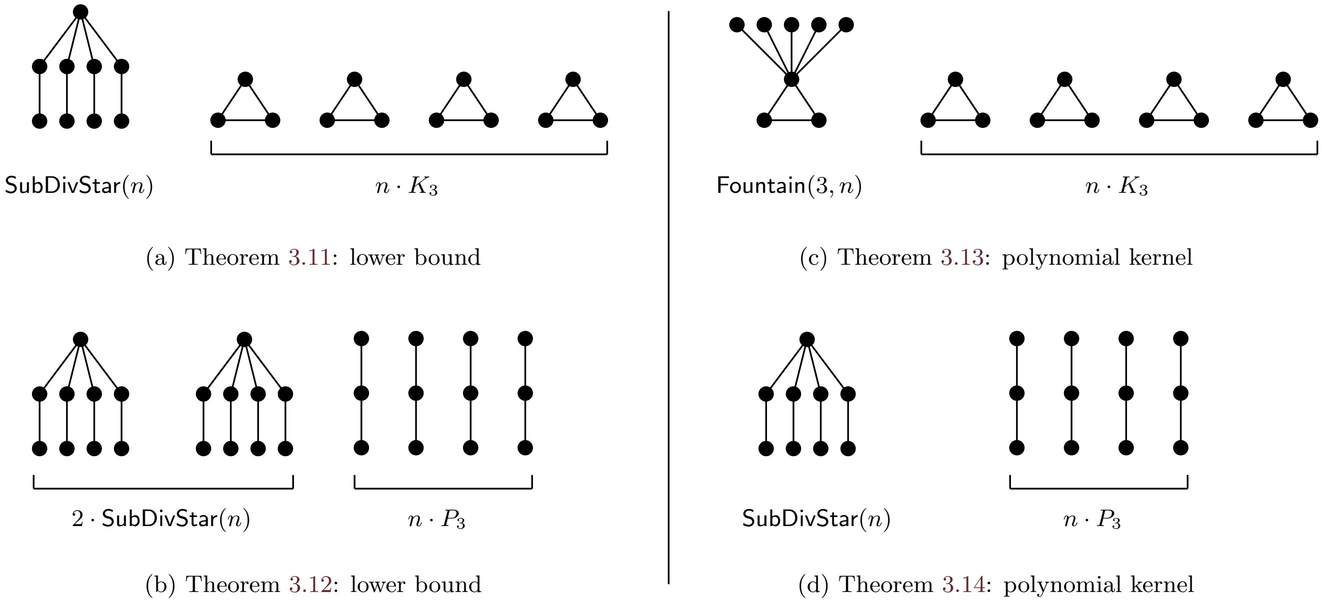

Figure 2: Illustrating the classes of graphs in Theorems 3.11–3.14.

3.5 Motivation for hereditary classes In this paper, we restricted our study to hereditary classes $\mathcal{F}$. There are a number of reasons motivating this decision. First, considering arbitrary classes $\mathcal{F}$ can make it very hard to prove lower bounds by polynomial-time reductions (even if the classes are decidable). For a concrete example, pick $\mathcal{F}$ consisting of every clique of size $2^{2^{2^i}}$ for $i \geq 1$. Then $\mathcal{F}$-Subgraph Test is unlikely to be polynomial-time solvable, but this seems difficult to prove with a polynomial-time reduction as the smallest clique in $\mathcal{F}$ of size exceeding n may have superpolynomial size. A difficulty of different sorts appears if $\mathcal{F}$ contains cliques such that the sizes of cliques in $\mathcal{F}$ form a more dense set of integers than in the previous example, but deciding if the clique of a particular size n is in $\mathcal{F}$ takes time exponential in n. These issues may be considered artifacts of trying to prove hardness by uniform polynomial-time reductions that work for every input length n; potentially the issues can be avoided by formulating the complexity framework in a different way. However, there are even more substantial difficulties that appear when the class $\mathcal{F}$ is not hereditary. For example, let $\mathcal{F}$ be the set of all paths. Then $\mathcal{F}$-Subgraph Test is NP-hard and it does not admit a polynomial kernel, unless NP $\subseteq$ coNP/poly [7]. Consider now the class $\mathcal{F}'$ containing, for every $i \geq 1$, the graph formed by a path of length i together with 2^i isolated vertices. The introduction of the isolated vertices should not change the complexity of the problem, but, surprisingly, it does. The problem of finding a path of length k in an n-vertex graph can be solved in time $2^{O(k)} \cdot n^{O(1)}$ [2, 50, 6]. Therefore, if H consists of a path of length k and 2^k isolated vertices, then these algorithms give a polynomial-time algorithm for finding H in a graph G: the running time $2^{O(k)} \cdot n^{O(1)}$ is polynomial in the size of H and G. Therefore, $\mathcal{F}'$-Subgraph Test is polynomial-time solvable, but apparently only because finding a path of length k is fixed-parameter tractable and has $2^{O(k)}n^{O(1)}$ time algorithms (note that the $2^{O(k \log k)}n^{O(1)}$ time algorithm of Monien [45] would not be sufficient for this argument). Therefore, it seems that we need a very tight understanding of the fixed-parameter tractability of $\mathcal{F}$-Subgraph Test to argue about its polynomial-time solvability. There are examples in the literature where the polynomial-time solvability of a problem was characterized for every (not necessarily hereditary) class $\mathcal{F}$ [13, 15, 27, 28], but in all these results, the characterization of polynomial-time was possible only because it coincided with fixed-parameter tractability. There is certainly no such coincidence for $\mathcal{F}$-Subgraph Test (for example, finding a path of length k is NP-hard, but FPT) and moreover the fixed-parameter tractability of $\mathcal{F}$-Subgraph Test is not well understood, as shown, for example, by the Biclique problem.

All these problems disappear if we restrict $\mathcal{F}$ to be hereditary (e.g., adding isolated vertices certainly cannot make the problem easier and if $\mathcal{F}$ contains arbitrary large cliques, then $\mathcal{F}$ contains every clique). While this restricts the generality of our results to some extent, we believe that avoiding the difficulties discussed above more than compensates for this lack of generality.

Let us also comment on the fact that, while the problems we study concern finding and packing (non-induced) subgraphs, we characterize the difficulty of such problems for each class $\mathcal{F}$ of pattern graphs that is closed under induced subgraphs (i.e., for hereditary classes). The discrepancy between induced and non-induced here is entirely natural. Note that every class $\mathcal{F}$ of pattern graphs that is closed under subgraphs, is also closed under induced subgraphs, and is therefore covered by our dichotomies. The fact that we can also classify $\mathcal{F}$ that are merely closed under induced subgraphs, rather than normal subgraphs, gives our results extra strength.

We mention in passing the classical result of Lewis and Yannakakis [40] on fully characterizing the complexity of vertex-deletion problems defined by hereditary properties; these results also rely crucially on the assumption that the property is hereditary. Note that our results on $\mathcal{F}$-Subgraph Test are unrelated to the results of Lewis and Yannakakis [40]: their problem is related to finding induced subgraphs and the task is not to find a specific induced subgraph, but to find a subgraph belonging to the class and having a specified size.

4 Conclusion

In this extended abstract we have presented our results on $\mathcal{F}$-Subgraph Test and $\mathcal{F}$-Packing and motivated them. Due to the great length of the proofs, they have been deferred to the full version of the paper, which is accessible on arXiv. We therefore invite interested readers to access the proofs online.

References

[1] F. N. Abu-Khzam. An improved kernelization algorithm for r-Set Packing. *Inf. Process. Lett.*, 110(16):621–624, 2010. doi:10.1016/j.ipl.2010.04.020.

[2] N. Alon, R. Yuster, and U. Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995.

[3] A. M. Ambalath, R. Balasundaram, C. R. H., V. Koppula, N. Misra, G. Philip, and M. S. Ramanujan. On the kernelization complexity of colorful motifs. In *Proc. 5th IPEC*, pages 14–25, 2010. doi:10.1007/978-3-642-17493-3_4.

[4] A. Atminas, V. V. Lozin, and I. Razgon. Linear time algorithm for computing a small biclique in graphs without long induced paths. In *Proc. 13th SWAT*, pages 142–152, 2012. doi:10.1007/978-3-642-31155-0_13.

[5] D. Binkele-Raible, H. Fernau, F. V. Fomin, D. Lokshtanov, S. Saurabh, and Y. Villanger. Kernel(s) for problems with no kernel: On out-trees with many leaves. *ACM Trans. Algorithms*, 8(4):38, 2012. doi:10.1145/2344422.2344428.

[6] A. Björklund. Determinant sums for undirected hamiltonicity. *SIAM J. Comput.*, 43(1):280–299, 2014. doi:10.1137/110839229.

[7] H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009. doi:10.1016/j.jcss.2009.04.001.

[8] H. L. Bodlaender, F. V. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh, and D. M. Thilikos. (Meta) Kernelization. In *Proc. 50th FOCS*, pages 629–638, 2009. doi:10.1109/FOCS.2009.46.

[9] H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Kernel bounds for path and cycle problems. *Theor. Comput. Sci.*, 511:117–136, 2013. arXiv:1106.4141, doi:10.1016/j.tcs.2012.09.006.

[10] H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Kernelization lower bounds by cross-composition. *SIAM J. Discrete Math.*, 28(1):277–305, 2014. arXiv:1206.5941, doi:10.1137/120880240.

[11] H. L. Bodlaender, S. Thomassé, and A. Yeo. Kernel bounds for disjoint cycles and disjoint paths. *Theor. Comput. Sci.*, 412(35):4570–4578, 2011. doi:10.1016/j.tcs.2011.04.039.

[12] J. Chen, S. Lu, S.-H. Sze, and F. Zhang. Improved algorithms for path, matching, and packing problems. In *Proc. 18th SODA*, pages 298–307, 2007. URL: http://doi.acm.org/10.1145/1283383.1283415.

[13] Y. Chen, M. Thurley, and M. Weyer. Understanding the complexity of induced subgraph isomorphisms. In *Proc. 35th ICALP*, pages 587–596, 2008. doi:10.1007/978-3-540-70575-8_48.

[14] M. Cygan, S. Kratsch, M. Pilipczuk, M. Pilipczuk, and M. Wahlström. Clique cover and graph separation: New incompressibility results. In *Proc. 39th ICALP*, pages 254–265, 2012. doi:10.1007/978-3-642-31594-7_22.

[15] V. Dalmau and P. Jonsson. The complexity of counting homomorphisms seen from the other side. *Theor. Comput. Sci.*, 329(1-3):315–323, 2004. doi:10.1016/j.tcs.2004.08.008.

[16] H. Dell and D. Marx. Kernelization of packing problems. In *Proc. 23rd SODA*, pages 68–81, 2012.

[17] H. Dell and D. van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In *Proc. 42nd STOC*, pages 251–260, 2010. doi:10.1145/1806689.1806725.

[18] M. Dom, D. Lokshtanov, and S. Saurabh. Incompressibility through colors and IDs. In *Proc. 36th ICALP*, pages 378–389, 2009. doi:10.1007/978-3-642-02927-1_32.

[19] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

[20] A. Drucker. New limits to classical and quantum instance compression. In *Proc. 53rd FOCS*, pages 609–

618, 2012. doi:10.1109/FOCS.2012.71.

[21] P. Erdős and R. Rado. Intersection theorems for systems of sets. *J. London Math. Soc.*, 35:85–90, 1960.

[22] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer-Verlag New York, Inc., 2006.

[23] F. V. Fomin, D. Lokshtanov, N. Misra, and S. Saurabh. Planar $\mathcal{F}$-Deletion: Approximation, kernelization and optimal FPT algorithms. In *Proc. 53rd FOCS*, pages 470–479, 2012. doi:10.1109/FOCS.2012.62.

[24] F. V. Fomin, D. Lokshtanov, V. Raman, S. Saurabh, and B. V. R. Rao. Faster algorithms for finding and counting subgraphs. *J. Comput. Syst. Sci.*, 78(3):698–706, 2012. doi:10.1016/j.jcss.2011.10.001.

[25] L. Fortnow and R. Santhanam. Infeasibility of instance compression and succinct PCPs for NP. *J. Comput. Syst. Sci.*, 77(1):91–106, 2011. doi:10.1016/j.jcss.2010.06.007.

[26] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.

[27] M. Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM*, 54(1):1, 2007. doi:10.1145/1206035.1206036.

[28] M. Grohe, T. Schwentick, and L. Segoufin. When is the evaluation of conjunctive queries tractable? In *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 657–666, New York, NY, USA, 2001. ACM Press. doi:http://doi.acm.org/10.1145/380752.380867.

[29] D. Hermelin, S. Kratsch, K. Sołtys, M. Wahlström, and X. Wu. A completeness theory for polynomial (Turing) kernelization. In *Proc. 8th IPEC*, pages 202–215, 2013. doi:10.1007/978-3-319-03898-8_18.

[30] D. Hermelin and X. Wu. Weak compositions and their applications to polynomial lower bounds for kernelization. In *Proc. 23rd SODA*, pages 104–113, 2012.

[31] B. M. P. Jansen. Turing kernelization for finding long paths and cycles in restricted graph classes. In *Proc. 22nd ESA*, pages 579–591, 2014. arXiv:1402.4718.

[32] B. M. P. Jansen and H. L. Bodlaender. Vertex cover kernelization revisited: Upper and lower bounds for a refined parameter. In *Proc. 28th STACS*, pages 177–188, 2011. doi:10.4230/LIPIcs.STACS.2011.177.

[33] B. M. P. Jansen and S. Kratsch. On polynomial kernels for structural parameterizations of odd cycle transversal. In *Proc. 6th IPEC*, pages 132–144, 2011. doi:10.1007/978-3-642-28050-4_11.

[34] B. M. P. Jansen and S. Kratsch. Data reduction for graph coloring problems. *Inform. Comput.*, 231:70–88, 2013. arXiv:1104.4229, doi:10.1016/j.ic.2013.08.005.

[35] S. Khot and V. Raman. Parameterized complexity of finding subgraphs with hereditary properties. *Theor. Comput. Sci.*, 289(2):997–1008, 2002. doi:10.1016/S0304-3975(01)00414-5.

[36] D. G. Kirkpatrick and P. Hell. On the completeness of a generalized matching problem. In *Proc. 10th STOC*, pages 240–245, 1978. doi:10.1145/800133.804353.

[37] J. Kneis, D. Mölle, S. Richter, and P. Rossmanith. Divide-and-color. In *Proc. 32nd WG*, pages 58–67, 2006. doi:10.1007/11917496_6.

[38] S. Kratsch. Co-nondeterminism in compositions: a kernelization lower bound for a Ramsey-type problem. In *ACM Trans. Algorithms*, 2013. To appear.

[39] S. Kratsch, M. Pilipczuk, A. Rai, and V. Raman. Kernel lower bounds using co-nondeterminism: Finding induced hereditary subgraphs. In *Proc. 13th SWAT*, pages 364–375, 2012. doi:10.1007/978-3-642-31155-0_32.

[40] J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *J. Comput. Syst. Sci.*, 20(2):219–230, 1980.

[41] B. Lin. The parameterized complexity of k-Biclique. In *Proc. 26th SODA*, 2014. In press. arXiv:1406.3700.

[42] D. Lokshtanov, N. Misra, and S. Saurabh. Kernelization - Preprocessing with a guarantee. In *The Multivariate Algorithmic Revolution and Beyond*, pages 129–161, 2012. doi:10.1007/978-3-642-30891-8_10.

[43] D. Marx. List edge multicoloring in graphs with few cycles. *Inf. Process. Lett.*, 89(2):85–90, 2004. doi:10.1016/j.ipl.2003.09.016.

[44] D. Marx and M. Pilipczuk. Everything you always wanted to know about the parameterized complexity of subgraph isomorphism (but were afraid to ask). In *Proc. 31st STACS*, pages 542–553, 2014. doi:10.4230/LIPIcs.STACS.2014.542.

[45] B. Monien. How to find long paths efficiently. *Annals of Discrete Mathematics*, 25:239–254, 1985. doi:10.1016/S0304-0208(08)73110-4.

[46] H. Moser. A problem kernelization for graph packing. In *Proc. 35th SOFSEM*, pages 401–412, 2009. doi:10.1007/978-3-540-95891-8_37.

[47] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987. doi:10.1007/BF02579206.

[48] A. Schäfer, C. Komusiewicz, H. Moser, and R. Niedermeier. Parameterized computational complexity of finding small-diameter subgraphs. *Optim. Lett.*, 6(5):883–891, 2012. doi:10.1007/s11590-011-0311-5.

[49] S. Thomassé, N. Trotignon, and K. Vuskovic. Parameterized algorithm for weighted independent set problem in bull-free graphs. arXiv 1310.6205, 2013. arXiv:1310.6205.

[50] R. Williams. Finding paths of length k in $O^*(2^k)$ time. *Inf. Process. Lett.*, 109(6):315–318, 2009. doi:10.1016/j.ipl.2008.11.004.

[51] V. V. Williams and R. Williams. Finding, minimizing, and counting weighted subgraphs. *SIAM J. Comput.*, 42(3):831–854, 2013. doi:10.1137/09076619X.

[52] M. Yannakakis. Node-deletion problems on bipartite graphs. *SIAM J. Comput.*, 10(2):310–327, 1981. doi:10.1137/0210022.

[53] C.-K. Yap. Some consequences of non-uniform conditions on uniform classes. *Theor. Comput. Sci.*, 26:287–

300, 1983. doi:10.1016/0304-3975(83)90020-8.