Linear kernels and single-exponential algorithms via protrusion decompositions *

Eun Jung Kim† Alexander Langer‡ Christophe Paul§ Felix Reidl‡
Peter Rossmanith‡ Ignasi Sau§ Somnath Sikdar‡

Abstract

We present a linear-time algorithm to compute a decomposition scheme for graphs G that have a set $X \subseteq V(G)$, called a *treewidth-modulator*, such that the treewidth of $G - X$ is bounded by a constant. Our decomposition, called a *protrusion decomposition*, is the cornerstone in obtaining the following two main results.

Our first result is that any parameterized graph problem (with parameter k) that has *finite integer index* and such that YES-instances have a treewidth-modulator of size $O(k)$ admits a linear kernel on the class of H-topological-minor-free graphs, for any fixed graph H. This result partially extends previous meta-theorems on the existence of linear kernels on graphs of bounded genus and H-minor-free graphs.

Let $\mathcal{F}$ be a fixed finite family of graphs containing at least one planar graph. Given an n-vertex graph G and a non-negative integer k, PLANAR-$\mathcal{F}$-DELETION asks whether G has a set $X \subseteq V(G)$ such that $|X| \leqslant k$ and $G - X$ is H-minor-free for every $H \in \mathcal{F}$. As our second application, we present the first *single-exponential* algorithm to solve PLANAR-$\mathcal{F}$-DELETION. Namely, our algorithm runs in time $2^{O(k)} \cdot n^2$, which is asymptotically optimal with respect to k. So far, single-exponential algorithms were only known for special cases of the family $\mathcal{F}$.

Keywords: parameterized complexity, algorithmic meta-theorems, sparse graphs, graph minors, hitting minors.

*We would like to point out that this article replaces and extends the results of [CoRR, abs/1201.2780, 2012]. Research funded by DFG-Project RO 927/12-1 "Theoretical and Practical Aspects of Kernelization", ANR project AGAPE (ANR-09-BLAN-0159), and the Languedoc-Roussillon Project "Chercheur d'avenir" KERNEL. A preliminary version of this work appeared in the proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP 2013), volume 7965 of LNCS.

†CNRS, LAMSADE, Paris, France, eunjungkim78@gmail.com.

‡Theoretical Computer Science, Department of Computer Science, RWTH Aachen University, Germany, {langer,reidl,rossmani,sikdar}@cs.rwth-aachen.de.

§CNRS, LIRMM, Montpellier, France, {paul,sau}@lirmm.fr.

1 Introduction

All graphs in this paper are finite undirected graphs. Parameterized complexity deals with algorithms for decision problems whose instances consist of a pair (x, k), where k is a secondary measurement of the input known as the *parameter*. A major goal in parameterized complexity is to investigate whether a problem with parameter k admits an algorithm with running time $f(k) \cdot |x|^{O(1)}$, where f is a function depending only on the parameter and $|x|$ represents the input size. Parameterized problems that admit such algorithms are called *fixed-parameter tractable* and the class of all such problems is denoted FPT. For an introduction to the area see [32, 35, 72].

A closely related concept is that of *kernelization*. A kernelization algorithm, or just *kernel*, for a parameterized problem Π takes an instance (x, k) of the problem and, in time polynomial in $|x| + k$, outputs an instance (x', k') such that $|x'|, k' \leqslant g(k)$ for some function g, and $(x, k) \in \Pi$ if and only if $(x', k') \in \Pi$. The function g is called the *size* of the kernel and may be viewed as a measure of the "compressibility" of a problem using polynomial-time preprocessing rules. It is a folklore result that a decidable problem is in FPT if and only if it has a kernelization algorithm. However, the kernel that one obtains in this way is typically of size at least exponential in the parameter. A natural problem in this context is to find polynomial or linear kernels for problems that are in FPT.

This work contributes to the two main areas of parameterized complexity mentioned above, namely, kernels and fixed-parameter tractable (FPT) algorithms. In many cases, the key ingredient in order to solve a hard graph problem is to find an appropriate *decomposition* of the input graph, which allows to take advantage of the structure given by the graph class and/or the problem under study. In this article we follow this paradigm and present a novel linear-time algorithm to compute a decomposition with nice properties for graphs G that have a set $X \subseteq V(G)$, called *t-treewidth-modulator*, such that the treewidth of $G - X$ is at most some constant $t - 1$. We then exploit this decomposition in two different ways: to *analyze* the size of kernels and to obtain *efficient* FPT algorithms. We would like to note that similar decompositions have already been (explicitly or implicitly) used for obtaining polynomial kernels [3, 8, 38, 42, 50].

Linear kernels. During the last decade, a plethora of results emerged on linear kernels for graph problems restricted to *sparse* graph classes. A celebrated result is the linear kernel for Dominating Set on planar graphs by Alber *et al.* [3]. This paper prompted an explosion of research papers on linear kernels on planar graphs, including Dominating Set [3, 19], Feedback Vertex Set [9], Cycle Packing [10], Induced Matching [54, 69], Full-Degree Spanning Tree [51], and Connected Dominating Set [64]. Guo and Niedermeier [50] designed a general framework and showed that problems that satisfy a certain "distance property" have linear kernels on planar graphs. This result was subsumed by that of Bodlaender *et al.* [8] who provided a meta-theorem for problems to have a linear kernel on graphs of bounded genus, a strictly larger class than planar graphs. Later Fomin *et al.* [42] extended these results for bidimensional problems to an even larger graph class, namely, H-minor-free and apex-minor-free graphs. (In all these works, the problems are parameterized by the *solution size*. See also [46, 47] for some recent meta-kernelization results considering structural parameters.) A common feature of these meta-theorems on sparse graphs is a *decomposition scheme* of the input graph that, loosely speaking, allows to deal with each part of the decomposition independently. For instance, the approach of [50], which is much inspired from [3], is to consider a so-called *region decomposition* of the input planar graph. The key point is that in an appropriately reduced Yes-instance, there are $O(k)$ regions and each one has constant size, yielding the desired linear kernel. This idea was generalized in [8] to graphs on surfaces, where the role of regions is played by *protrusions*, which are graphs with small treewidth and small boundary (see Section 2 for details). The resulting decomposition is called *protrusion decomposition*. A crucial point is that while the reduction rules of [3] are *problem-dependent*, those of [8] are *automated*, relying on a property called *finite integer index* (FII), which was introduced by Bodlaender and de Fluiter [12]. Loosely speaking (see Section 2), having FII guarantees that "large" protrusions of a graph can be replaced by "small" gadget graphs preserving equivalence of instances. This operation is usually called the *protrusion replacement rule*. FII is also of central importance to the approach of [42] on H-minor-free graphs. In fact, the idea of protrusion replacement (using a different terminology) can be traced back to the early 90's in the work of Arnborg *et al.* [4], and afterwards Bodlaender and de Fluiter [12] generalized the results in [4] to optimization problems. See also [**?**] for related work on this area.

Following the spirit of the aforementioned results, we present a novel algorithm to compute protrusion decompositions that allows us to obtain linear kernels on a larger class of sparse graphs, namely H-topological-minor-free graphs. Our algorithm takes as input a graph G and a t-treewidth-modulator $X \subseteq V(G)$, and outputs a set of vertices Y_0 containing X such that every connected component of $G - Y_0$ is a protrusion (see Section 3 for details).

When G is the input graph of a parameterized graph problem Π with parameter k, we call a protrusion decomposition of G *linear* if both $|Y_0|$ and the number of protrusions of $G - Y_0$ are $O(k)$. If Π is such that YES-instances have a t-treewidth-modulator of size $O(k)$ for some constant t (such problems are called *treewidth-bounding*, see Section 4), and G excludes some fixed graph H as a topological minor, we prove that the protrusion decomposition given by our algorithm is linear. If in addition Π has FII, then each protrusion can be replaced with a gadget of constant size, obtaining an equivalent instance of size $O(k)$. Our first main result summarizes the above discussion.

Theorem I. *Fix a graph H. Let Π be a parameterized graph problem on the class of H-topological-minor-free graphs that is treewidth-bounding and has finite integer index. Then Π admits a linear kernel.*

Consequences of Theorem I. It turns out that a host of problems including TREEWIDTH-t VERTEX DELETION, CHORDAL VERTEX DELETION, INTERVAL VERTEX DELETION, EDGE DOMINATING SET, to name a few, satisfy the conditions of our theorem. Since for any fixed graph H, the class of H-topological-minor-free graphs strictly contains the class of H-minor-free graphs, our result is in fact an extension of the results of Fomin *et al.* [42]. As we discuss in Section 6, there is evidence that our result may reach the limit of sparse graph classes for which there exist meta-theorems about the existence of linear (or even uniform polynomial) kernels.

We also exemplify how our algorithm to obtain a linear protrusion decomposition can be applied to obtain *explicit* linear kernels, that is, kernels without using a generic protrusion replacement. This is shown by exhibiting a simple explicit linear kernel for the EDGE DOMINATING SET problem on H-topological-minor-free graphs. So far, all known linear kernels for EDGE DOMINATING SET on H-minor-free graphs [42] and H-topological-minor-free graphs (given by Theorem I) relied on generic protrusion replacement.

Single-exponential algorithms. In order to prove Theorem I, similarly to [8, 42, 50] our protrusion decomposition algorithm is only used to *analyze* the size of the resulting instance after having applied the protrusion reduction rule. In the second part of the paper we show that our decomposition scheme can also be used to obtain *efficient* FPT algorithms. Before stating our second main result, let us motivate the problem that we study.

During the last decades, parameterized complexity theory has brought forth several algorithmic meta-theorems that imply that a wide range of problems are in FPT (see [59] for a survey). For instance, Courcelle's theorem [21] states that every decision problem expressible in Monadic Second Order Logic can be solved in linear time when parameterized by the treewidth of the input graph. At the price of generality, such algorithmic meta-theorems may suffer from the fact that the function $f(k)$ is huge [45,60] or non-explicit [21,80]. Therefore, it has become a central task in parameterized complexity to provide FPT algorithms such that the behavior of the function $f(k)$ is *reasonable*; in other words, a function $f(k)$ that could lead to a practical algorithm.

Towards this goal, we say that an FPT parameterized problem is solvable in *single-exponential* time if there exists an algorithm solving it in time $2^{O(k)} \cdot n^{O(1)}$. For instance, recent results have shown that broad families of problems admit (deterministic or randomized) single-exponential algorithms parameterized by treewidth [23,31,82]. On the other hand, single-exponential algorithms are unlikely to exist for certain parameterized problems [23,63]. Parameterizing by the size of the desired solution, in the case of Vertex Cover the existence of a single-exponential algorithm has been known for a long time, but it took a while to witness the first (deterministic) single-exponential algorithm for Feedback Vertex Set, or equivalently Treewidth-One Vertex Deletion [27,49].

Both Vertex Cover and Feedback Vertex Set can be seen as graph modification problems in order to attain a hereditary property, that is, a property closed under taking induced subgraphs. It is well-known that deciding whether at most k vertices can be deleted from a given graph in order to attain any non-trivial hereditary property is NP-complete [61]. The particular case where the property can be characterized by a finite set of forbidden induced subgraphs can be solved in single-exponential time when parameterizing by the number of modifications, even in the more general case where also edge deletions or additions are allowed [16]. If the family of forbidden induced subgraphs is infinite, no meta-theorem is known and not every problem is even FPT [62]. A natural question arises: can we carve out a larger class of hereditary properties for which the corresponding graph modification problem can be solved in single-exponential time?

A line of research emerged pursuing this question, which is much inspired by the Feedback Vertex Set problem. Interestingly, when the *infinite* family of forbidden *induced subgraphs* can also be captured by a *finite* set $\mathcal{F}$ of forbidden *minors* (namely, when the problem is closed under taking minors [81]), the $\mathcal{F}$-Deletion problem (namely, the problem of removing at most k vertices from an input graph to obtain a graph which is H-minor-free for every $H \in \mathcal{F}$) is in non-uniform[1] FPT by the seminal meta-theorem of Robertson and Seymour [80][2].

Let $\mathcal{F}$ be a finite family of graphs containing at least one planar graph. The parameterized problem that we consider in the second part of the paper is Planar-$\mathcal{F}$-Deletion: given a graph G and a non-negative integer parameter k as input, does G have a set $X \subseteq V(G)$ such that $|X| \leqslant k$ and $G - X$ is H-minor-free for every $H \in \mathcal{F}$?

Planar-$\mathcal{F}$-Deletion

[1]A non-uniform FPT algorithm for a parameterized problem is a collection of algorithms, one for each value of the parameter k.

[2]It is worth noting that, in contrast to the removal of vertices, the problems corresponding to the operations of removing or contracting edges are not minor-closed (we provide a proof of this fact in Appendix A), and therefore the result of Robertson and Seymour [80] cannot be applied to these modification problems.

Input:	A graph G and a non-negative integer k.
Parameter:	The integer k.
Question:	Does G have a set $X \subseteq V(G)$ such that $\|X\| \leqslant k$ and $G - X$ is H-minor-free for every $H \in \mathcal{F}$?

Note that Vertex Cover and Feedback Vertex Set correspond to the special cases of $\mathcal{F} = \{K_2\}$ and $\mathcal{F} = \{K_3\}$, respectively. A recent work by Joret *et al.* [53] handled the case $\mathcal{F} = \{\theta_c\}$ and achieved a single-exponential algorithm for Planar-θ_c-Deletion for any value of $c \geqslant 1$, where θ_c is the (multi)graph consisting of two vertices and c parallel edges between them. (Note that the cases $c = 1$ and $c = 2$ correspond to Vertex Cover and Feedback Vertex Set, respectively.) Kim *et al.* [56] obtained a single-exponential algorithm for $\mathcal{F} = \{K_4\}$, also known as Treewidth-Two Vertex Deletion. Related works of Philip *et al.* [74] and Cygan *et al.* [24] resolve the case $\mathcal{F} = \{K_3, T_2\}$, or equivalently Pathwidth-One Vertex Deletion, in single-exponential time.

The Planar-$\mathcal{F}$-Deletion problem was first stated by Fellows and Langston [33], who proposed a non-uniform (and non-constructive) $f(k) \cdot n^2$-time algorithm for some function $f(k)$, as well as a $f(k) \cdot n^3$-time algorithm for the general $\mathcal{F}$-Deletion problem, both relying on the meta-theorem of Robertson and Seymour [80]. Explicit bounds on the function $f(k)$ for Planar-$\mathcal{F}$-Deletion can be obtained via dynamic programming. Indeed, as the Yes-instances of Planar-$\mathcal{F}$-Deletion have treewidth $O(k)$, using standard dynamic programming techniques on graphs of bounded treewidth (see for instance [2,6]), it can be seen that Planar-$\mathcal{F}$-Deletion can be solved in time $f(k) \cdot n^2$ with $f(k) = 2^{2^{O(k \log k)}}$. In a recent unpublished paper [39], Fomin *et al.* proposed a $2^{O(k \log k)} \cdot n^2$-time algorithm for Planar-$\mathcal{F}$-Deletion, which is, up to our knowledge, the best known result. More recently, Fomin *et al.* [40] provided a $2^{O(k)} \cdot n \log^2 n$-time algorithm for the Planar-Connected-$\mathcal{F}$-Deletion problem, which is the special case of Planar-$\mathcal{F}$-Deletion when *every* graph in the family $\mathcal{F}$ is *connected*. In this paper, we get rid of the connectivity assumption, and we prove that the general Planar-$\mathcal{F}$-Deletion problem can be solved in single-exponential time. Namely, our second main result is the following.

Theorem II. *The parameterized* Planar-$\mathcal{F}$-Deletion *problem can be solved in time* $2^{O(k)} \cdot n^2$.

This result unifies, generalizes, and simplifies a number of results given in [20, 27, 40, 49, 53, 56]. Let us make a few considerations about the fact that the family $\mathcal{F}$ may contain disconnected graphs or not. Besides the fact that removing the connectivity constraint is an important theoretical step towards the general $\mathcal{F}$-Deletion problem, it turns out that many natural such families $\mathcal{F}$ do contain disconnected graphs. For instance, the disjoint union of g copies of K_5 (or $K_{3,3}$) is a minimal forbidden minor, or an *obstruction*, for the graphs of genus $g - 1$ [5] (see also [68]). In particular, the (disconnected) graph made of two copies of K_5 is in the obstruction set of the graphs that can be embedded in the torus. In Appendix B we show that many natural obstruction sets also contain disconnected *planar* graphs.

It should also be noted that the function $2^{O(k)}$ in Theorem II is best possible, assuming the Exponential Time Hypothesis[3] (ETH). Namely, it is known that unless the ETH fails, Vertex Cover cannot be solved in time $2^{o(k)} \cdot poly(n)$ [35, Chapter 16]. It is noteworthy that when $\mathcal{F}$ does not contain any planar graph, up to our knowledge no single case is known to admit a single-exponential algorithm. For instance, we point out that Planar Vertex Deletion, which amounts to $\mathcal{F} = \{K_5, K_{3,3}\}$, is not known to have a single-exponential parameterized algorithm [67], while a double-exponential function $f(k)$ is the best known so far [55].

[3] ETH states that there exists $\delta > 0$ such that 3-SAT does not allow an algorithm running in time $O(2^{\delta n} \cdot poly(n))$.

Technical ingredients in the proof of Theorem II. As mentioned above, when employing protrusion replacement, often the problem needs to have FII. Many problems enjoy this property, for example Treewidth-t Vertex Deletion or (Connected) Dominating Set, among others. Having FII makes the problem amenable to this powerful reduction rule, and essentially this was the basic ingredient of previous works such as [40, 53, 56]. In particular, when every graph in $\mathcal{F}$ is connected, the Planar-$\mathcal{F}$-Deletion problem has FII [8], and the single-exponential time algorithm of [40] heavily depends on this feature. However, if one aims at Planar-$\mathcal{F}$-Deletion without any connectivity restriction on the family $\mathcal{F}$, the requirement for FII seems to be a fundamental hurdle, as if $\mathcal{F}$ may contain disconnected graphs, then Planar-$\mathcal{F}$-Deletion does not have FII for some choices of $\mathcal{F}$.[4] We observe that the unpublished $2^{O(k \log k)} \cdot n^2$-time algorithm of [39] applies to the general Planar-$\mathcal{F}$-Deletion problem (that is, $\mathcal{F}$ may contain some disconnected graph). The reason is that instead of relying on FII, they rather use tools from *annotated kernelization* [8].

To circumvent the situation of not having FII, our algorithm *does not use any reduction rule*, but instead relies on a series of branching steps. First of all, we apply the iterative compression technique (introduced by Reed *et al.* [76]) in order to reduce the Planar-$\mathcal{F}$-Deletion problem to its *disjoint* version. In the Disjoint Planar-$\mathcal{F}$-Deletion problem, given a graph G and an initial solution X of size k, the task is to decide whether G contains an alternative solution $\tilde{X}$ disjoint from X of size at most $k-1$. In our case, the assumption that $\mathcal{F}$ contains some planar graph is fundamental, as then $G - X$ has bounded treewidth [78]. Central to our single-exponential algorithm is our linear-time algorithm to compute a protrusion decomposition, in this case with the initial solution X as treewidth-modulator. A first step to this end is to use the aforementioned algorithm (the one used for the analysis of linear kernel) and compute a superset Y_0 of X such that each component of $G - Y_0$, together with its neighborhood in Y_0, forms a protrusion. But for the resulting protrusion decomposition to be linear, it turns out that we first need to guess the intersection of the alternative solution with the set Y_0. Once we have the desired linear protrusion decomposition, instead of applying protrusion replacement, we simply identify a set of $O(k)$ vertices among which the alternative solution has to live, if it exists. In the whole process described above, there are three branching steps: the first one is inherent to the iterative compression paradigm, the second one is required to compute a linear protrusion-decomposition, and finally the last one enables us to guess the set of vertices containing the solution. It can be proved that each branching step is compatible with single-exponential time, which yields the desired result. Finally, it is worth mentioning that our algorithm is *fully constructive* (cf. Section 5.3 for details).

Organization of the paper. In Section 2, we outline all important definitions that are relevant to this work. We then exhibit our protrusion decomposition algorithm in Section 3. As our first application of our decomposition result, we prove Theorem I in Section 4. In Section 5 we prove Theorem II. Finally, in Section 6 we conclude with some closing remarks.

2 Preliminaries

We use standard graph-theoretic notation (see [28] for any undefined terminology). Given a graph G, we let $V(G)$ denote its vertex set and $E(G)$ its edge set. For convenience we assume that $V(G)$ is a totally ordered set. The neighborhood of a vertex $x \in V(G)$ is the set of all vertices $y \in V(G)$

[4] As we were not able to find a reference with a proof of this fact, for completeness we provide it in Appendix C.

such that $xy \in E(G)$ and is denoted by $N^G(x)$. The closed neighborhood of x is defined as $N^G[x] := N^G(x) \cup \{x\}$. The distance $d_G(x,y)$ of two vertices $x, y \in V(G)$ is the length (number of edges) of a shortest x,y-path in G and ∞ if x, y lie in different connected components of G. The *rth neighborhood* of a vertex $N_r^G(v) := \{w \in G \mid d_G(v,w) \leqslant r\}$ is the set of vertices within distance at most r to v, in particular we have that $N_0^G(v) = \{v\}$ and $N_1^G(v) = N^G[v]$. Since we will mainly be concerned with sparse graphs in this paper, we let $|G|$ denote the number of vertices in the graph G. Subscripts and superscripts are omitted if it is clear which graph is being referred to. For $X \subseteq V(G)$, we let $G[X]$ denote the graph (X, E_X), where $E_X := \{xy \mid x, y \in X \text{ and } xy \in E(G)\}$, and we define $G - X := G[V(G) \setminus X]$.

By the *neighbors of a subgraph* $H \subseteq G$, denoted $N^G(H)$, we mean the set of vertices in $V(G) \setminus V(H)$ that have at least one neighbor in H. We employ the same notation analogously to denote *neighbors of a subset of vertices* $N^G(S)$ for $S \subseteq V(G)$. If X is a subset of vertices disjoint from S, then $N_X^G(S)$ is the set $N^G(S) \cap X$. The same notation naturally extends to a subgraph $H \subseteq G$, that is, $N_X^G(H)$. (Again, when the graph G is clear from the context, we may drop it from the notation.) We denote by $\omega(G)$ the size of the largest complete subgraph of G and by $\#\omega(G)$ the *number* of complete subgraphs (not necessarily maximal ones). Given an edge $e = xy$ of a graph G, we let G/e denote the graph obtained from G by *contracting* the edge e, which amounts to deleting the endpoints of e, introducing a new vertex v_{xy}, and making it adjacent to all vertices in $(N^G(x) \cup N^G(y)) \setminus \{x, y\}$. A *minor* of G is a graph obtained from a subgraph of G by contracting zero or more edges. If H is a minor of G, we write $H \preceq_m G$. A graph G is *H-minor-free* if $H \npreceq_m G$. A *topological minor* of G is a graph obtained from a subgraph of G by contracting zero or more edges, such that each contracted edge has at least one endpoint with degree at most two. We write $H \preceq_{tm} G$ to denote that H is a topological minor of G. Note that $H \preceq_{tm} G$ implies that $H \preceq_m G$, but not vice-versa. A graph G is *H-topological-minor-free* if $H \npreceq_{tm} G$.

2.1 Parameterized problems, kernels and treewidth

A parameterized problem Π is a subset of $\Gamma^* \times \mathbb{N}_0$, where Γ is some finite alphabet. An instance of a parameterized problem is a tuple (x, k), where k is the parameter.

Definition 1 (Parameterized graph problem)**.** A *parameterized graph problem* Π is a subset of $\{(G, k) \mid G \text{ is a graph and } k \in \mathbb{N}_0\}$. If $\mathcal{G}$ is a graph class, we define the problem Π *restricted to* $\mathcal{G}$ as $\Pi_\mathcal{G} = \{(G,k) \mid (G,k) \in \Pi \text{ and } G \in \mathcal{G}\}$.

A parameterized problem Π is *fixed-parameter tractable* (FPT for short) if there exists an algorithm that decides instances (x, k) in time $f(k) \cdot poly(|x|)$, where f is a function of k alone. The notion of kernelization is defined as follows.

Definition 2 (Kernelization)**.** A *kernelization algorithm*, or just *kernel*, for a parameterized problem $\Pi \subseteq \Gamma^* \times \mathbb{N}_0$ is an algorithm that given $(x, k) \in \Gamma^* \times \mathbb{N}_0$ outputs, in time polynomial in $|x| + k$, an instance $(x', k') \in \Gamma^* \times \mathbb{N}_0$ such that:

1. $(x, k) \in \Pi$ if and only if $(x', k') \in \Pi$;
2. $|x'|, k' \leqslant g(k)$,

where g is some computable function. The function g is called the *size* of the kernel. If $g(k) = k^{O(1)}$ or $g(k) = O(k)$, we say that Π admits a *polynomial kernel* or a *linear kernel*, respectively.

Definition 3 (Treewidth)**.** Given a graph $G = (V, E)$, a *tree-decomposition of* G is an ordered pair $(T, \{W_x \mid x \in V(T)\})$, where T is a tree, such that the following hold:

1. $\bigcup_{x \in V(T)} W_x = V(G)$;
2. for every edge $e = uv$ in G, there exists $x \in V(T)$ such that $u, v \in W_x$;
3. for each vertex $u \in V(G)$, the set of nodes $\{x \in V(T) \mid u \in W_x\}$ induces a subtree of T.

The vertices of the tree T are usually referred to as *nodes* and the sets W_x are called *bags*. The *width* of a tree-decomposition is the size of a largest bag minus one. The *treewidth* of G, denoted $\mathbf{tw}(G)$, is the smallest width of a tree-decomposition of G.

Given a bag B of a tree-decomposition with a rooted tree T, we denote by T_B the subtree rooted at the node corresponding to bag B, and by $G_B := G[\bigcup_{x \in T_B} W_x]$ the subgraph of G induced by the vertices appearing in the bags corresponding to the nodes of T_B. A *join bag* B of a rooted tree-decomposition is a bag such that the root of T_B has degree at least two. If a graph G is disconnected, a *forest-decomposition* of G is the union of tree-decompositions of its connected components. We refer the reader to Diestel's book [28] for an introduction to the theory of treewidth. For the definition of *nice tree-decomposition*, we refer the readers to [57].

2.2 (Counting) Monadic Second Order Logic

Monadic Second Order Logic (MSO) is an extension of First Order Logic that allows quantification over sets of objects. We identify graphs with relational structures over a vocabulary τ_{Graph}, consisting of the unary relation symbols $\textsc{Vert}$ and $\textsc{Edge}$ and the binary relation symbol $\textsc{Inc}$. A graph $G = (V, E)$ is then represented by a τ_{Graph}-structure $\mathcal{G}$ with universe $U(\mathcal{G}) = V \cup E$ such that:

- $\textsc{Vert}^{\mathcal{G}} = V$ and $\textsc{Edge}^{\mathcal{G}} = E$ represent the vertex set and the edge set, respectively, and
- $\textsc{Inc}^{\mathcal{G}} = \{\, (v, e) \mid v \in V, e \in E$ and v is incident to $e \,\}$ represents the incidence relation.

A *Monadic Second Order* formula contains two types of variables: *individual variables* to be used for elements of the universe, usually denoted by lowercase letters $x, y, z, \ldots$ and *set variables* to be used for subsets of the universe, usually denoted by uppercase letters $X, Y, Z, \ldots$. Atomic formulas on τ_{Graph} are: $x = y$, $x \in X$, $x \in \textsc{Vert}$, $x \in \textsc{Edge}$, and $\textsc{Inc}(x, y)$ for all individual variables x, y and set variables X. MSO formulas on τ_{Graph} are built from the atomic formulas using Boolean connectives $\neg, \wedge, \vee$, and quantification $\exists x, \forall x, \forall X, \forall Y$ for individual variables x and set variables X. MSO formulas are interpreted in τ_{Graph}-structures in the natural way, e.g., $\textsc{Inc}(x, y)$ being true iff in G the vertex v represented by x is incident to the edge e represented by y.

In a *Counting* Monadic Second Order (CMSO) formula, we have additional atomic formulas $\mathbf{card}_{n,p}(X)$ on set variables X, which are true if the set U represented by the variable X has size n (mod p). We refer to [22, 35] for a more detailed presentation on (C)MSO logic. In a p-$\textsc{min}$-CMSO graph problem (respectively, p-$\textsc{max}$-CMSO or p-$\textsc{eq}$-CMSO) Π, one has to decide the existence of a set S of at most k vertices/edges (respectively, at least k or exactly k) in an input graph G such that the CMSO expressible predicate $P_\Pi(G, S)$ is satisfied.

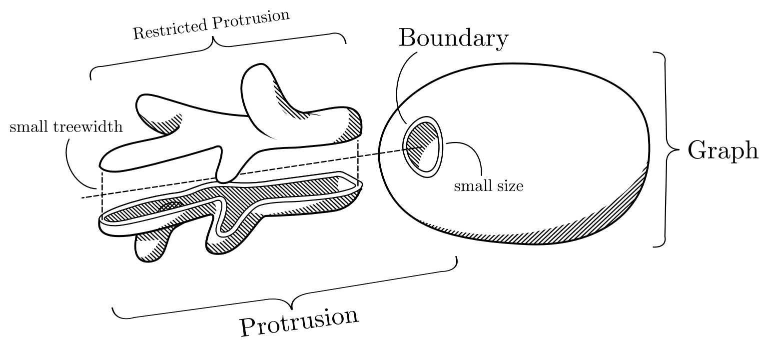

Figure 1: Basic anatomy of a protrusion.

2.3 Protrusions, t-boundaried graphs, and finite integer index

We restate the main definitions of the protrusion machinery developed in [8,42]. Given a graph $G = (V, E)$ and a set $W \subseteq V$, we define $\partial_G(W)$ as the set of vertices in W that have a neighbor in $V \setminus W$. For a set $W \subseteq V$ the neighborhood of W is $N^G(W) = \partial_G(V \setminus W)$. Superscripts and subscripts are omitted when it is clear which graph is being referred to.

Definition 4 (t-protrusion [8])**.** Given a graph G, a set $W \subseteq V(G)$ is a *t-protrusion* of G if $|\partial_G(W)| \leqslant t$ and $\mathbf{tw}(G[W]) \leqslant t - 1$.[5] If W is a t-protrusion, the vertex set $W' = W \setminus \partial_G(W)$ is the *restricted protrusion of* W. We call $\partial_G(W)$ the *boundary* and $|W|$ the *size* of the t-protrusion W of G. Given a restricted t-protrusion W', we denote its *extended protrusion* by $W'^{+} = W' \cup N(W')$.

Note that if W' is the restricted protrusion of W, then $W'^{+} = W$. A rough outline of a protrusion is depicted in Figure 1.

A *t-boundaried graph* is a graph $G = (V, E)$ with a set $\mathbf{bd}(G)$ (called the *boundary*[6] or the *terminals* of G) of t distinguished vertices labeled 1 through t. Let $\mathcal{G}_t$ denote the class of t-boundaried graphs, with graphs from $\mathcal{G}$. If $W \subseteq V$ is an r-protrusion in G, then we let G_W be the r-boundaried graph $G[W]$ with boundary $\partial_G(W)$, where the vertices of $\partial_G(W)$ are assigned labels 1 through r according to their order in G.

Definition 5 (Gluing and ungluing)**.** For t-boundaried graphs G_1 and G_2, we let $G_1 \oplus G_2$ denote the graph obtained by taking the disjoint union of G_1 and G_2 and identifying each vertex in $\mathbf{bd}(G_1)$ with the vertex in $\mathbf{bd}(G_2)$ with the same label. This operation is called *gluing*.

Let $G_1 \subseteq G$ with a boundary B of size t. The operation of *ungluing* G_1 from G creates the t-boundaried graph $G \ominus_B G_1 := G - (V(G_1) \setminus B)$ with boundary B. The vertices of $\mathbf{bd}(G \ominus_B G_1)$ are assigned labels 1 through t according to their order in the graph G.

Note that the gluing operation entails taking the union of edges both of whose endpoints are in the boundary with the deletion of multiple edges to keep the graph simple. The ungluing operation preserves the boundary (both the vertices and the edges).

Definition 6 (Replacement)**.** Let $G = (V, E)$ be a graph with a t-protrusion W; let G_W denote the graph $G[W]$ with boundary $\mathbf{bd}(G_W) = \partial_G(W)$; and finally, let G_1 be a t-boundaried graph. Then *replacing* G_W by G_1 corresponds to the operation $(G \ominus G_W) \oplus G_1$.

[5] In [8], $\mathbf{tw}(G[W]) \leqslant t$, but we want the size of the bags to be at most t.

[6] Usually denoted by $\partial(G)$, but this collides with our usage of ∂.

Definition 7 (Protrusion decomposition)**.** An *(α, t)-protrusion decomposition* of a graph G is a partition $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ of $V(G)$ such that:

1. for every $1 \leqslant i \leqslant \ell$, $N(Y_i) \subseteq Y_0$;
2. $\max\{\ell, |Y_0|\} \leqslant \alpha$;
3. for every $1 \leqslant i \leqslant \ell$, $Y_i \cup N_{Y_0}(Y_i)$ is a t-protrusion of G.

The set Y_0 is called the *separating part* of $\mathcal{P}$.

Hereafter, the value of t will be fixed to some constant. When G is the input of a parameterized graph problem with parameter k, we say that an (α, t)-protrusion decomposition of G is *linear* (resp. *quadratic*) whenever $\alpha = O(k)$ (resp. $\alpha = O(k^2)$).

We now restate the definition of one of the most important notions used in this paper.

Definition 8 (Finite integer index (FII) [12])**.** Let $\Pi_{\mathcal{G}}$ be a parameterized graph problem restricted to a class $\mathcal{G}$ and let G_1, G_2 be two t-boundaried graphs in $\mathcal{G}_t$. We say that $G_1 \equiv_{\Pi,t} G_2$ if there exists a constant $\Delta_{\Pi,t}(G_1, G_2)$ (that depends on Π, t, and the ordered pair (G_1, G_2)) such that for all t-boundaried graphs G_3 and for all k:

1. $G_1 \oplus G_3 \in \mathcal{G}$ iff $G_2 \oplus G_3 \in \mathcal{G}$;
2. $(G_1 \oplus G_3, k) \in \Pi$ iff $(G_2 \oplus G_3, k + \Delta_{\Pi,t}(G_1, G_2)) \in \Pi$.

We say that the problem $\Pi_{\mathcal{G}}$ has *finite integer index in the class* $\mathcal{G}$ iff for every integer t, the equivalence relation $\equiv_{\Pi,t}$ has finite index (that is, it has a finite number of equivalence classes). In the case that $(G_1 \oplus G, k) \notin \Pi$ or $G_1 \oplus G \notin \mathcal{G}$ for *all* $G \in \mathcal{G}_t$, we set $\Delta_{\Pi,t}(G_1, G_2) = 0$. Note that $\Delta_{\Pi,t}(G_1, G_2) = -\Delta_{\Pi,t}(G_2, G_1)$.

We would like to note that the definition of finite integer index given in Definition 8 differs from the definition given in [8, 12], where the first condition in the definition of the equivalence relation $\equiv_{\Pi,t}$ is not required. As in [42], we adopt the above definition for notational simplicity, due to the following reason. Assume that the equivalence relation defined only by the second condition in Definition 8 has finite index. Assume furthermore that the membership in the graph class $\mathcal{G}$ can be expressed in MSO logic. Then by Myhill-Nerode's theorem[7], it follows that the equivalence relation $\equiv_{\Pi,t}$ has finite index as well (see for instance [32, 35, 72]). As the the membership in the class of graphs that can exclude a fixed graph H as a topological minor can be expressed in MSO logic (see Appendix D for the precise formula), it will be simpler to already incorporate the first condition in Definition 8.

If a parameterized problem has finite integer index then its instances can be reduced by "replacing protrusions". The technique of replacing protrusions hinges on the fact that each protrusion of "large" size can be replaced by a "small" gadget from the same equivalence class as the protrusion, which consequently behaves similarly w.r.t. the problem at hand. If G_1 is replaced by a gadget G_2, then the parameter k in the problem changes by $\Delta_{\Pi,t}(G_1, G_2)$. What is not immediately clear is that given that a problem Π has finite integer index, how does one show that there always exists a set of representatives for which the parameter is guaranteed not to *increase*. The next lemma shows that this is indeed the case. The ideas of the proof are implicit in [12].

[7]A thorough historical study of the family of Myhill-Nerode's theorems can be found in [84].

Lemma 1. *Let* Π *be a parameterized graph problem that has finite integer index in a graph class* $\mathcal{G}$*. Then for every fixed* t*, there exists a finite set* $\mathcal{R}_t$ *of* t*-boundaried graphs such that for each* t*-boundaried graph* $G \in \mathcal{G}_t$ *there exists a* t*-boundaried graph* $G' \in \mathcal{R}_t$ *such that* $G \equiv_{\Pi,t} G'$ *and* $\Delta_{\Pi,t}(G, G') \geqslant 0$*.*

Proof. The set $\mathcal{R}_t$ consists of one element from each equivalence class of $\equiv_{\Pi,t}$. Since Π has finite integer index, the set $\mathcal{R}_t$ is finite. Therefore we only have to show that there exist representatives that satisfy the requirement in the statement of the lemma.

To this end, fix any equivalence class $\mathcal{G}'_t \in \mathcal{G}_t/{\equiv_{\Pi,t}}$. First consider the case where there exists $G_1 \in \mathcal{G}'_t$ such that for all $G \in \mathcal{G}_t$, either $G_1 \oplus G \notin \mathcal{G}$ or for all $k \in \mathbb{N}_0$, $(G_1 \oplus G, k) \notin \Pi$. Since $\mathcal{G}'_t$ is an equivalence class, this means that at least one of these two conditions holds for *every* graph $G \in \mathcal{G}'_t$. Thus $\Delta_{\Pi,t}(G_1, G_2) = 0$ for all t-boundaried graphs $G_1, G_2 \in \mathcal{G}'_t$ and we can simply take a graph of smallest size from $\mathcal{G}'_t$ as representative.

We can now assume that for the chosen $\mathcal{G}'_t$ it holds that there exists a t-boundaried graph $G \in \mathcal{G}_t$ such that for *all* $G_1 \in \mathcal{G}'_t$ we have that $G_1 \oplus G \in \mathcal{G}$ and, for some $k \in \mathbb{N}$, $(G_1 \oplus G, k) \in \Pi_{\mathcal{G}}$. Consider the following binary relation $\preceq$ over $\mathcal{G}'_t$: for all $G_1, G_2 \in \mathcal{G}'_t$,

$$G_1 \preceq G_2 \Leftrightarrow \Delta_{\Pi,t}(G_1, G_2) \geqslant 0.$$

As $\Delta_{\Pi,t}(G, G) = 0$ for all $G \in \mathcal{G}_t$, it immediately follows that the relation $\preceq$ is reflexive. Furthermore, the relation is *total* as every graph is comparable to every other graph from the same equivalence class.

We next show that the relation $\preceq$ is also transitive, making it a total quasi-order. Let $G_1, G_2, G_3 \in \mathcal{G}'_t$ be such that $G_1 \preceq G_2$ and $G_2 \preceq G_3$. This is equivalent to saying that $c_{12} = \Delta_{\Pi,t}(G_1, G_2) \geqslant 0$ and $c_{23} = \Delta_{\Pi,t}(G_2, G_3) \geqslant 0$. For every $G \in \mathcal{G}_t$ such that $G_1 \oplus G \in \mathcal{G}$ and $(G_1 \oplus G, k) \in \Pi$ for some $k \in \mathbb{N}$, we have

$$\begin{aligned}(G_1 \oplus G, k) \in \Pi \quad &\Leftrightarrow \quad (G_2 \oplus G, k + c_{12}) \in \Pi \\ &\Leftrightarrow \quad (G_3 \oplus G, k + c_{12} + c_{23}) \in \Pi.\end{aligned}$$

By definition, $\Delta_{\Pi,t}(G_1, G_3) = c_{12} + c_{23} \geqslant 0$ and hence $G_1 \preceq G_3$. We conclude that $\preceq$ is transitive and therefore a total quasi-order.

We now show that the class $\mathcal{G}'_t$ can be partitioned into layers that can be linearly ordered. We will pick our representative for the class $\mathcal{G}'_t$ from the first layer in this ordering. To do this, we define the following equivalence relation over $\mathcal{G}'_t$. For all $G_1, G_2 \in \mathcal{G}'_t$, define

$$\begin{aligned}G_1 \equiv G_2 \quad &\Leftrightarrow \quad G_1 \preceq G_2 \text{ and } G_2 \preceq G_1 \\ &\Leftrightarrow \quad \Delta_{\Pi,t}(G_1, G_2) = 0.\end{aligned}$$

Now, the equivalence classes $\mathcal{G}'_t/{\equiv}$ can be linearly ordered as follows. Fix a graph $G \in \mathcal{G}_t$ such that for any $G_1 \in \mathcal{G}'_t$ we have that $G_1 \oplus G \in \mathcal{G}$ and $(G_1 \oplus G, k) \in \Pi$ for some $k \in \mathbb{N}$, this graph must exist since we handled equivalence classes of $\mathcal{G}_t/{\equiv_{\Pi,t}}$ which do not have such a graph in the first part of the proof. Consider the function $\Phi_G \colon \mathcal{G}'_t/{\equiv} \to \mathbb{N}_0$ defined via

$$\Phi_G([G']) = \min \left\{ k \in \mathbb{N} \mid (G' \oplus G, k) \in \Pi \right\}.$$

Observe that $\Phi_G([G_2]) = \Phi_G([G_1]) + \Delta_{\Pi,t}(G_1, G_2)$ for all $G_1, G_2 \in \mathcal{G}'_t$ and, in particular, that

$$\Phi_G([G_1]) = \Phi_G([G_2]) \Leftrightarrow G_1 \equiv G_2.$$

Thus Φ_G induces a linear order on $\mathcal{G}'_t/{\equiv}$. Moreover, since $\Phi_G(\cdot) \geqslant 0$, there exists a class $[G^*]$ in $\mathcal{G}'_t/{\equiv}$ that is a minimum element in the order induced by Φ_G. For any t-boundaried graph $G \in [G^*]$, it then follows that for all $G_1 \in \mathcal{G}'_t$, $\Delta_{\Pi,t}(G, G_1) \geqslant 0$. The representative of $\mathcal{G}'_t$ in $\mathcal{R}_t$ is an arbitrary t-boundaried graph $G' \in [G^*]$ of smallest size. This proves the lemma. □

Definition 9 (Protrusion limit)**.** For a parameterized graph problem Π that has finite integer index in the class $\mathcal{G}$, let $\mathcal{R}_t$ denote the set of representatives of the equivalence classes of $\equiv_{\Pi,t}$ satisfying the requirement stated in Lemma 1. The *protrusion limit* of $\Pi_{\mathcal{G}}$ is defined as $\rho_{\Pi_{\mathcal{G}}}(t) = \max_{G \in \mathcal{R}_t} |V(G)|$. We drop the subscript when it is clear which graph problem is being referred to. We also define $\rho'(t) := \rho(2t)$.

The next two lemmas deal with finding protrusions in graphs. The first of these guarantees that whenever there exists a "large enough" protrusion there exists a protrusion that is large but of size bounded by a *constant* (that depends on the problem and the boundary size). As we shall see later, the fact that we deal with protrusions of constant size enables us to efficiently test which representative to replace them by, assuming that we have the set of representatives. For completeness, we provide the proof of the following lemma.

Lemma 2 ([8])**.** *Let Π be a parameterized graph problem with finite integer index in $\mathcal{G}$ and let $t \in \mathbb{N}$ be a constant. For a graph $G \in \mathcal{G}$, if one is given a t-protrusion $X \subseteq V(G)$ such that $\rho'_{\Pi_{\mathcal{G}}}(t) < |X|$, then one can, in time $O(|X|)$, find a $2t$-protrusion W such that $\rho'_{\Pi_{\mathcal{G}}}(t) < |W| \leqslant 2 \cdot \rho'_{\Pi_{\mathcal{G}}}(t)$.*

Proof. Let $(T, \mathcal{X})$ be a nice tree-decomposition for $G[X]$ of width $t-1$. Root T at an arbitrary node. Let u be the *lowest* node of T such that if W is the set of vertices in the bags associated with the nodes in the subtree T_u rooted at u, then $|W| > \rho'_{\Pi_{\mathcal{G}}}(t)$. Clearly W is a $2t$-protrusion with boundary $X_u \cup \partial_G(X)$, where $X_u \subseteq V(G)$ is the bag associated with the node u of T. By the choice of u, it is clear that u cannot be a forget node. If u is an introduce node with child v, then the number of vertices in the bags associated with the nodes of T_v must be exactly $\rho'_{\Pi_{\mathcal{G}}}(t)$. Since u introduces exactly one additional vertex of G, we have $|W| = \rho'_{\Pi_{\mathcal{G}}}(t) + 1$. Finally consider the case when u is a join node with children y, z. Then the bags associated with these nodes X_u, X_y, X_z are identical and since

$$|\bigcup_{j \in V(T_y)} X_j\,| < \rho'_{\Pi_{\mathcal{G}}}(t) \qquad \text{and} \qquad |\bigcup_{j \in V(T_z)} X_j\,| < \rho'_{\Pi_{\mathcal{G}}}(t),$$

we have that $W = \bigcup_{j \in V(T_y)} X_j \cup \bigcup_{j \in V(T_z)} X_j$ has size at most $2 \cdot \rho'_{\Pi_{\mathcal{G}}}(t)$.

Computing a nice tree-decomposition $(T, \mathcal{X})$ of $G[X]$ takes time $2^{O(t^3)} \cdot |X|$ [7] and the time required to compute a $2t$-protrusion from T is $O(|X|)$. Since t is a constant, the total time taken is $O(|X|)$. □

For a fixed t, the protrusion W is of *constant* size but, in the reduction rule to be described, would be replaced by a representative of smaller size, namely at most $\rho'_{\Pi_{\mathcal{G}}}(t) = \rho_{\Pi_{\mathcal{G}}}(2t)$. This means that each time the reduction rule is applied, the size of the graph strictly decreases and, by Lemma 1, the parameter does not increase. The reduction rule can therefore be applied at most n times, where n is the number of vertices in the input graph. As we shall see later, each application of the reduction rule takes time polynomial in n, assuming that we are given the set of representatives. Therefore, in polynomial time, we would obtain an instance in which every t-protrusion has size at most $\rho_{\Pi_{\mathcal{G}}}(2t)$. This trick is described in [8] but is stated here for the sake of completeness.

The next lemma describes how to find a t-protrusion of maximum size.

Lemma 3 (Finding maximum sized protrusions)**.** *Let t be a constant. Given an n-vertex graph G, a t-protrusion of G with the maximum number of vertices can be found in time $O(n^{t+1})$.*

Proof. For a vertex set $B \subseteq V(G)$ of size at most t, let $C_{B,1}, \ldots, C_{B,p}$ be the connected components of $G - B$ such that, for $1 \leqslant i \leqslant p$, $\mathbf{tw}(G[V(C_{B,i}) \cup B]) \leqslant t$. The connected components of $G - B$ can be determined in $O(n)$ time and one can test whether the graph induced by $V(C_{B,i}) \cup B$ has treewidth at most $t-1$ in time $2^{O(t^3)} \cdot n$ [7]. Since we have assumed that t is a fixed constant, deciding whether the treewidth is within $t-1$ can be done in linear time. By definition, $\bigcup_{i=1}^{p} V(C_{B,i}) \cup B$ is a t-protrusion with boundary B. Conversely every t-protrusion W consists of a boundary $\partial(W)$ of size at most t such that the restricted protrusion $W' = W \setminus \partial(W)$ is a collection of connected components C of $G - \partial(W)$ satisfying the condition $\mathbf{tw}(G[V(C) \cup \partial(W)]) \leqslant t-1$. Therefore to find a t-protrusion of maximum size, one simply runs through all vertex sets B of size at most t and for each set determines the maximum t-protrusion with boundary B. The largest t-protrusion over all choices of the boundary B is a largest t-protrusion in the graph. All of this takes time $O(n^{t+1})$. $\square$

Finally, given a $2t$-protrusion W with the desired size constraints, we show how to determine which representative of our equivalence class is equivalent to $G[W]$.

Lemma 4. *Let Π be a parameterized graph problem that has finite integer index on $\mathcal{G}$. For a constant $t \in \mathbb{N}$, suppose that the set $\mathcal{R}_t$ of representatives of the equivalence relation $\equiv_{\Pi,t}$ is given. If W is a t-protrusion of size at most a fixed constant c, then one can decide in constant time which $G' \in \mathcal{R}_t$ satisfies $G' \equiv_{\Pi,t} G[W]$.*

Proof. Fix $G' \in \mathcal{R}_t$. We wish to test whether $G' \equiv_{\Pi,t} G[W]$. For each $\tilde{G} \in \mathcal{R}_t$, solve the problem Π on the constant-sized instances $G[W] \oplus \tilde{G}$ and $G' \oplus \tilde{G}$ and let $s(G[W], \tilde{G})$ and $s(G', \tilde{G})$ denote the value of the parameter associated with the problem. Then by the definition of finite integer index, we have $G' \equiv_{\Pi,t} G[W]$ if and only if $s(G[W], \tilde{G}) - s(G', \tilde{G})$ is the *same* for all $\tilde{G} \in \mathcal{R}_t$. To find out which graph in $\mathcal{R}_t$ is the correct representative of $G[W]$, we run this test for each graph in $\mathcal{R}_t$, of which there are a constant number. The total time taken is, therefore, a constant. $\square$

3 Constructing protrusion decompositions

In this section we present our algorithm to compute protrusion decompositions. Our approach is based on an algorithm which marks the bags of a tree-decomposition of an input graph G that comes equipped with a subset $X \subseteq V(G)$ such that the graph $G - X$ has bounded treewidth. Let henceforth t be an integer such that $\mathbf{tw}(G - X) \leqslant t - 1$ and let r be an integer that is also given to the algorithm. This parameter r will depend on the particular graph class to which G belongs and the precise problem one might want to solve (see Sections 4 and 5 for more details). More precisely, given optimal tree-decompositions of the connected components of $G - X$ with at least r neighbors in X, the bag marking algorithm greedily identifies a set $\mathcal{M}$ of bags in a bottom-up manner. The set $V(\mathcal{M})$ of vertices contained in marked bags together with X will form the separating part Y_0 of the protrusion decomposition. Bags will be marked in two different steps, called "Large-subgraph" and "LCA" marking steps. The bags marked in the Large-subgraph marking step will be mapped bijectively into a collection of pairwise vertex-disjoint connected subgraphs of $G - X$, each of which has a large neighborhood in X (namely, of size greater than r), implying in several particular cases a limited number of marked bags (see Sections 4 and 5). In order to guarantee that the connected components of $G - (X \cup V(\mathcal{M}))$ form protrusions with small boundary, in the LCA marking step the set $\mathcal{M}$ is closed under taking LCA's (least common ancestors; see Lemma 6. We would like to note that this technique was also used in [8, 38]). The precise description of the procedure can be found in Algorithm 1 below and a sketch of the decomposition is depicted in Figure 2.

Algorithm 1: BAG MARKING ALGORITHM

Input: A graph G, a subset $X \subseteq V(G)$ such that $\mathbf{tw}(G - X) \leqslant t - 1$, and an integer $r > 0$.

Set $\mathcal{M} \leftarrow \emptyset$ as the set of marked bags;
Compute an optimal rooted tree-decomposition $\mathcal{T}_C = (T_C, \mathcal{B}_C)$ of every connected component C of $G - X$ such that $|N_X(C)| \geqslant r$;

Repeat the following loop for every rooted tree-decomposition $\mathcal{T}_C$;
while *$\mathcal{T}_C$ contains an unprocessed bag* **do**

- Let B be an unprocessed bag at the farthest distance from the root of $\mathcal{T}_C$;
- **[LCA marking step]**
  if *B is the LCA of two bags of $\mathcal{M}$* **then**
  - $\mathcal{M} \leftarrow \mathcal{M} \cup \{B\}$ and remove the vertices of B from every bag of $\mathcal{T}_C$;
- **[Large-subgraph marking step]**
  else if *G_B contains a connected component C_B such that $|N_X(C_B)| \geqslant r$* **then**
  - $\mathcal{M} \leftarrow \mathcal{M} \cup \{B\}$ and remove the vertices of B from every bag of $\mathcal{T}_C$;
- Bag B is now processed;

return $Y_0 = X \cup V(\mathcal{M})$;

Before we discuss properties of the set $\mathcal{M}$ of marked bags and the set $Y_0 = X \cup V(\mathcal{M})$, let us establish the time complexity of the bag marking algorithm and describe how the Large-subgraph marking step can be implemented using dynamic programming techniques. Since the dynamic programming procedure is quite standard, we just sketch the main ideas.

Implementation and time complexity of Algorithm 1. First, an optimal tree-decomposition of every connected component C of $G - X$ such that $|N_X(C)| \geqslant r$ can be computed in time linear in $n = |V(G)|$ using the algorithm of Bodlaender for graphs of bounded treewidth [7]. We root such tree-decomposition at an arbitrary bag. For the sake of simplicity of the analysis, we can assume that the tree-decompositions are nice, but it is not necessary for the algorithm.

Note that the LCA marking step can clearly be performed in linear time. Let us now briefly discuss how we can detect, in the Large-subgraph marking step, if a graph G_B contains a connected component C_B such that $|N_X(C_B)| \geqslant r$ using dynamic programming. For each bag B of the tree-decomposition, we have to keep track of which vertices of B belong to the same connected component of G_B.

Note that we only need to remember the connected components of the graph G_B which intersect B, as the other ones will never be connected to the rest of the graph. For each such connected component C_B intersecting B, we also store $N_X(C_B)$, and note that by definition of the algorithm, it follows that for non-marked bags B, $|N_X(C_B)| < r$. At a "join" bag J with children B_1 and B_2, we merge the connected components of G_{B_1} and G_{B_2} sharing at least one vertex (which is necessarily in J), and update their neighborhood in X accordingly. If for some of these newly

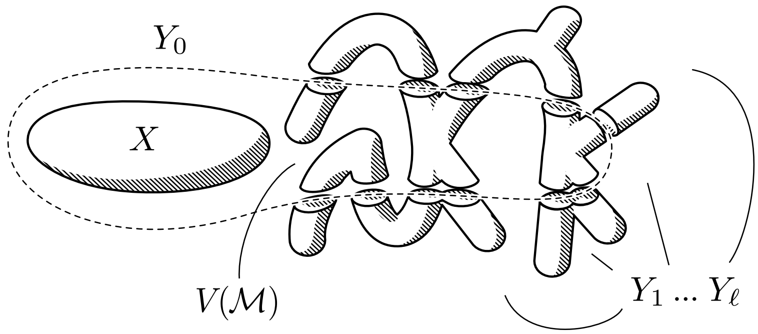

Figure 2: A sketch of how the marking algorithm obtains a protrusion decomposition. X denotes a treewidth-modulator. Edges among the individual vertex sets are not depicted.

created connected components C_J of G_J, it holds that $|N_X(C_J)| \geqslant r$, then the bag J needs to be marked. At a "forget" bag F corresponding to a forgotten vertex v, we only have to forget the connected component C of G_F containing v if $V(C) \cap F = \emptyset$. Finally, at an "introduce" bag I corresponding to a new vertex v, we have to merge connected components of G_I after the addition of vertex v, and update the neighbors in X according to the neighbors of v in X.

Note that for each bag B, the time needed to update the information about the connected components of G_B depends polynomially on t and r. In order for the whole algorithm to run in linear time, we can deal with the removal of marked vertices in the following way. Instead of removing them from every bag of the tree-decomposition, we can just label them as "marked" when marking a bag B, and just not take them into account when processing further bags.

The next lemma follows from the above discussion.

Lemma 5. *Algorithm 1 can be implemented to run in $O(n)$ time, where the hidden constant depends only on t and r.*

Basic properties of Algorithm 1. Denote by $\mathcal{T}$ the union of the set of optimal tree-decompositions $\mathcal{T}_C$ of every connected component C of $G - X$ with at least r neighbors in X.

Lemma 6. *If T is a maximal connected subtree of $\mathcal{T}$ not containing any marked bag of $\mathcal{M}$, then T is adjacent to at most two marked bags of $\mathcal{T}$.*

Proof. As every tree-decomposition in $\mathcal{T}$ is rooted, so is any maximal subtree T of $\mathcal{T}$ not containing any marked bag of $\mathcal{M}$. Assume that $\mathcal{M}$ contains two distinct marked bags, say B_1 and B_2, each adjacent to a leaf of T. As T is connected, observe that the LCA B of B_1 and B_2 belongs to T. Since $\mathcal{M}$ is closed under taking LCA, T contains a marked bag B, a contradiction. It follows that T is adjacent to at most two marked bags: a unique one adjacent to a leaf, and possibly another one adjacent to its root. □

As a consequence of the previous lemma we can now argue that every connected component of $G - Y_0$ has a small neighborhood in X and thus forms a restricted protrusion.

Lemma 7. *Let Y_0 be the set of vertices computed by Algorithm 1. Every connected component C of $G - Y_0$ satisfies $|N_X(C)| < r$ and $|N_{Y_0}(C)| < r + 2t$.*

Proof. Let C be a connected component of $G - Y_0$. Observe that C is contained in a connected component C_X of $G - X$ such that either $|N_X(C_X)| < r$ or $|N_X(C_X)| \geqslant r$. In the former case, as Algorithm 1 does not mark any vertex of C_X, $C = C_X$ and so $|N_{Y_0}(C)| < r + 2t$ trivially holds. So assume that $|N_X(C_X)| \geqslant r$. Then C_X has been chopped by Algorithm 1 and clearly $C \subseteq C_X \setminus V(\mathcal{M})$. More precisely, if $\mathcal{T}_{C_X}$ is the rooted tree-decomposition of C_X, there exists a maximal connected subtree T of $\mathcal{T}_{C_X}$ not containing any marked bag such that $C \subseteq V(T) \setminus V(\mathcal{M})$. By construction of $\mathcal{M}$, every connected component of the subgraph induced by $V(T) \setminus V(\mathcal{M})$ has strictly less than r neighbors in X (otherwise the root of T or one of its descendants would have been marked at the Large-subgraph marking step). It follows that $|N_X(C)| < r$. To conclude, observe that Lemma 6 implies that the neighbors of C in $V(\mathcal{M})$ are contained in at most two marked bags of $\mathcal{T}$. It follows that $|N_{Y_0}(C)| < r + 2t$. □

Given a graph G and a subset $S \subseteq V(G)$, we define a *cluster of* $G - S$ as a maximal collection of connected components of $G - S$ with the same neighborhood in S. Note that the set of all clusters of $G - S$ induces a partition of the set of connected components of $G - S$, which can be easily found in linear time if G and S are given.

By Lemma 7 and using the fact that $\mathbf{tw}(G - X) \leqslant t - 1$, the following proposition follows.

Proposition 1. *Let r, t be two positive integers, let G be a graph and $X \subseteq V(G)$ such that $\mathbf{tw}(G - X) \leqslant t - 1$, let $Y_0 \subseteq V(G)$ be the output of Algorithm 1 with input (G, X, r), and let $Y_1, \ldots, Y_\ell$ be the set of all clusters of $G - Y_0$. Then $\mathcal{P} := Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ is a $(\max\{\ell, |Y_0|\}, 2t + r)$-protrusion decomposition of G.*

In other words, each cluster of $G - Y_0$ is a restricted $(2t + r)$-protrusion. Note that Proposition 1 neither bounds ℓ nor $|Y_0|$. In the sequel, we will use Algorithm 1 and Proposition 1 to give explicit bounds on ℓ and $|Y_0|$, in order to achieve two different results. In Section 4 we use Algorithm 1 and Proposition 1 to obtain *linear kernels* for a large class of problems on *sparse* graphs. In Section 5 we use Algorithm 1 and Proposition 1 to obtain a *single-exponential algorithm* for the parameterized Planar-$\mathcal{F}$-Deletion problem.

4 Linear kernels on graphs excluding a topological minor

We start this section by proving Theorem I in Subsection 4.1. We then state a number of concrete problems that satisfy the structural constraints imposed by this theorem (Subsection 4.2), discuss these constraints in the context of previous work in this area (Subsection 4.3), and trace graph classes to which our approach can be lifted (Subsection 4.4). Finally (Subsection 4.5), we discuss how to use the machinery developed in proving Theorem I to obtain a concrete kernel for the Edge Dominating Set problem.

4.1 Proof of Theorem I

With the protrusion machinery outlined in Section 2 at hand, we can now describe the protrusion reduction rule. Informally, we find a sufficiently large t-protrusion (for some yet to be fixed constant t), replace it with a small representative, and change the parameter accordingly. In the following, we will drop the subscript from the protrusion limit functions ρ_Π and ρ'_Π.

Reduction Rule 1 (Protrusion reduction rule)**.** Let $\Pi_{\mathcal{G}}$ denote a parameterized graph problem restricted to some graph class $\mathcal{G}$, let $(G,k) \in \Pi_{\mathcal{G}}$ be a YES-instance of $\Pi_{\mathcal{G}}$, and let $t \in \mathbb{N}$ be a constant. Suppose that $W' \subseteq V(G)$ is a t-protrusion of G such that $|W'| > \rho'(t)$, obtained as described in Lemma 3. Let $W \subseteq V(G)$ be a $2t$-protrusion of G such that $\rho'(t) < |W| \leqslant 2 \cdot \rho'(t)$, obtained as described in Lemma 2. We let G_W denote the $2t$-boundaried graph $G[W]$ with boundary $\mathbf{bd}(G_W) = \partial_G(W)$. Let further $G_1 \in \mathcal{R}_{2t}$ be the representative of G_W for the equivalence relation $\equiv_{\Pi,|\partial_G(W)|}$ as defined in Lemma 1.
The protrusion reduction rule (for boundary size t) is the following:

Reduce (G,k) *to* $(G',k') = (G \ominus G_W \oplus G_1, k - \Delta_{\Pi,2t}(G_1, G_W))$.

By Lemma 1, the parameter in the new instance does not increase. We now show that the protrusion reduction rule is safe.

Lemma 8 (Safety)**.** *Let $\mathcal{G}$ be a graph class and let $\Pi_{\mathcal{G}}$ be a parameterized graph problem with finite integer index with respect to $\mathcal{G}$. If (G',k') is the instance obtained from one application of the protrusion reduction rule to the instance (G,k) of $\Pi_{\mathcal{G}}$, then*

1. *$G' \in \mathcal{G}$;*
2. *(G',k') is a* YES*-instance iff (G,k) is a* YES*-instance; and*
3. *$k' \leqslant k$.*

Proof. Suppose that (G',k') is obtained from (G,k) by replacing a $2t$-boundaried subgraph G_W (induced by a $2t$-protrusion W) by a representative $G_1 \in \mathcal{R}_{2t}$. Let $\tilde{G}$ be the $2t$-boundaried graph $G - W'$, where W' is the restricted protrusion of W and $\mathbf{bd}(\tilde{G}) = \partial_G(W)$. Since $G_W \equiv_{\Pi,2t} G_1$, we have by Definition 8,

1. $G = \tilde{G} \oplus G_W \in \mathcal{G}$ iff $\tilde{G} \oplus G_1 \in \mathcal{G}$.
2. $(\tilde{G} \oplus G_W, k) \in \Pi_{\mathcal{G}}$ iff $(\tilde{G} \oplus G_1, k - \Delta_{\Pi,2t}(G_1, G_W)) \in \Pi_{\mathcal{G}}$.

Hence $G' = \tilde{G} \oplus G_1 \in \mathcal{G}$. Lemma 1 ensures that $\Delta_{\Pi,2t}(G_1, G_W) \geqslant 0$, and hence $k' = k - \Delta_{\Pi,2t}(G_1, G_W)) \leqslant k$. □

Observation 1. *If (G,k) is reduced with respect to the protrusion reduction rule with boundary size β, then for all $t \leqslant \beta$, every t-protrusion W of G has size at most $\rho'(t)$.*

In order to obtain linear kernels, we require the problem instances to have more structure. In particular, we adapt the notion of *quasi-compactness* introduced in [8] to define what we call *treewidth-bounding.*

Definition 10 (Treewidth-bounding)**.** A parameterized graph problem $\Pi_{\mathcal{G}}$ is called (s,t)-*treewidth-bounding* if there exists a function $s\colon \mathbb{N} \to \mathbb{N}$ and a constant t such that for every $(G,k) \in \Pi_{\mathcal{G}}$ there exists $X \subseteq V(G)$ such that:

1. $|X| \leqslant s(k)$; and
2. $\mathbf{tw}(G - X) \leqslant t - 1$.

We call a problem *treewidth-bounding on a graph class* $\mathcal{G}$ if the above property holds under the restriction that $G \in \mathcal{G}$. We call X a *t-treewidth-modulator* of G, s the *treewidth-modulator size* and t the *treewidth bound* of the problem Π.

We assume in the following that the problem $\Pi_{\mathcal{G}}$ at hand is (s,t)-treewidth-bounding with bound t and modulator size $s(\cdot)$, that is, a Yes-instance $(G,k) \in \Pi_{\mathcal{G}}$ has a modulator set $X \subseteq V(G)$ with $|X| \leqslant s(k)$ and $\mathbf{tw}(G - X) \leqslant t - 1$. Note that in general s, t depend on $\Pi_{\mathcal{G}}$ and $\mathcal{G}$. For many problems that are treewidth-bounding, such as Vertex Cover, Feedback Vertex Set, Treewidth-t Vertex Deletion, the set X is actually the solution set. However, in general, X could be *any* vertex set and does not have to be given nor efficiently computable to obtain a kernel. The fact that it exists is all we need for our proof to go through.

The rough idea of the proof of Theorem I is as follows. We assume that the given instance (G,k) is reduced w.r.t. the protrusion reduction rule for some yet to be fixed constant boundary size β. Consequently, every β-protrusion of G has size at most $\rho'(\beta)$. For a protrusion decomposition $Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ obtained from Algorithm 1 with a carefully chosen threshold r (see Proposition 1), we can then show that $|Y_0| = O(k)$ using properties of H-topological-minor-free graphs. The bound on the total size of the clusters of $G - Y_0$ follows from these properties and from the protrusion reduction rule.

We first prove a result (Theorem 1) that is slightly more general than Theorem I and identifies all the key ingredients needed for our result. To do this, we use a sequence of lemmas (9, 10, 11) which bounds the total size of the clusters of the protrusion decomposition. To this end, we define the *constriction* operation, which essentially shrinks paths into edges.

Definition 11 (Constriction)**.** Let G be a graph and let $\mathcal{P}$ be a set of paths in G such that for each $P \in \mathcal{P}$ it holds that:

1. the endpoints of P are not connected by an edge in G; and
2. for all $P' \in \mathcal{P}$, with $P' \neq P$, P and P' share at most a single vertex which must also be an endpoint of both paths.

We define the *constriction* of G under $\mathcal{P}$, written $G|_{\mathcal{P}}$, as the graph H obtained by connecting the endpoints of each $P \in \mathcal{P}$ by an edge and then removing all inner vertices of P.

We say that H is a *d-constriction* of G if there exists $G' \subseteq G$ and a set of paths $\mathcal{P}$ in G' such that $d = \max_{P \in \mathcal{P}} |P|$ and $H = G'|_{\mathcal{P}}$. Given graph classes $\mathcal{G}, \mathcal{H}$ and some integer $d \geqslant 2$, we say that $\mathcal{G}$ *d-constricts into* $\mathcal{H}$ if for every $G \in \mathcal{G}$, every possible d-constriction H of G is contained in the class $\mathcal{H}$. For the case that $\mathcal{G} = \mathcal{H}$ we say that $\mathcal{G}$ is *closed under d-constrictions*. We will call $\mathcal{H}$ the *witness* class, as the proof of Theorem 1 works by taking an input graph G and constricting it into some witness graph H whose properties will yield the desired bound on $|G|$. We let $\omega(G)$ denote the size of a largest clique in G and $\#\omega(G)$ the total number of cliques in G (not necessarily maximal ones).

Theorem 1. *Let $\mathcal{G}, \mathcal{H}$ be graph classes closed under taking subgraphs such that $\mathcal{G}$ d-constricts into $\mathcal{H}$ for a fixed constant $d \in \mathbb{N}$. Assume that $\mathcal{H}$ has the property that there exist functions $f_E, f_{\#\omega} \colon \mathbb{N} \to \mathbb{N}$ and a constant $\omega_{\mathcal{H}}$ (depending only on $\mathcal{H}$) such that for each graph $H \in \mathcal{H}$ the following conditions hold:*

$$|E(H)| \leqslant f_E(|H|), \ \ \#\omega(H) \leqslant f_{\#\omega}(|H|), \ \ \textit{and } \omega(H) < \omega_{\mathcal{H}}.$$

Let Π be a parameterized graph problem that has finite integer index and is (s,t)-treewidth-bounding, both on the graph class $\mathcal{G}$. Define $x_k := s(k) + 2t \cdot f_E(s(k))$. Then any reduced instance $(G,k) \in \Pi$ has a protrusion decomposition $V(G) = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ such that:

1. $|Y_0| \leqslant x_k$*;*
2. $|Y_i| \leqslant \rho'(2t + \omega_{\mathcal{H}})$ *for* $1 \leqslant i \leqslant \ell$*; and*
3. $\ell \leqslant f_{\#\omega}(x_k) + f_E(x_k) + x_k + 1$.

Hence Π restricted to $\mathcal{G}$ admits kernels of size at most

$$x_k + (f_{\#\omega}(x_k) + f_E(x_k) + x_k + 1)\rho'(2t + \omega_{\mathcal{H}}).$$

Even if it is not our main objective to optimize the running time of the kernelization algorithm given by Theorem 1, we just note that it is dominated by the algorithm of Lemma 3 for finding protrusions. We split the proof of Theorem 1 into several lemmas. First, let us fix the way in which the decomposition $Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ is obtained: given a reduced YES-instance $(G,k) \in \Pi$, let $X \subseteq V(G)$ be a treewidth-modulator of size at most $|X| \leqslant s(k)$ such that $\mathbf{tw}(G - X) \leqslant t - 1$. For the analysis of the kernel size, we run Algorithm 1 on the input $(G, X, \omega_{\mathcal{H}})$.

Lemma 9. *The protrusion decomposition $Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ obtained by running Algorithm 1 on $(G, X, \omega_{\mathcal{H}})$ has the following properties:*

1. *For each $1 \leqslant i \leqslant \ell$, we have $|Y_i| \leqslant \rho'(2t + \omega_{\mathcal{H}})$;*
2. *For each connected subgraph C_B found by Algorithm 1 in the "Large-subgraph marking step", $|C_B| \leqslant \rho'(2t + \omega_{\mathcal{H}}) + t$.*

Proof. The first claim follows directly from Lemma 7: for each $1 \leqslant i \leqslant \ell$, we have $|N_{Y_0}(Y_i)| \leqslant 2t + \omega_{\mathcal{H}}$. As $Y_i \subseteq G - X$, it follows that $\mathbf{tw}(G[Y_i]) \leqslant t - 1$ and therefore Y_i forms a restricted $(2t + \omega_{\mathcal{H}})$-protrusion in G. Since our instance is reduced, we have $|Y_i| \leqslant \rho'(2t + \omega_{\mathcal{H}})$.

Note that during a run of the algorithm, if a bag B currently being considered is not marked, then each connected component C_B of G_B satisfies $|N_X(C_B)| < \omega_{\mathcal{H}}$. Hence C_B along with its neighbors in X is a $(2t+\omega_{\mathcal{H}})$-protrusion and since the instance is reduced we have $|C_B| \leqslant \rho'(2t+\omega_{\mathcal{H}})$. Moreover the algorithm ensures that $|N_R(C_B)| \leqslant 2t$, where $R = V(G) \setminus X$, and thus a component with a neighborhood larger than $2t + \omega_{\mathcal{H}}$ must have at least $\omega_{\mathcal{H}}$ neighbors in X. Now as every step of the algorithm adds at most t more vertices to the components of G_B, it follows that once a component with at least $\omega_{\mathcal{H}}$ neighbors in X *is* found, it can contain at most $\rho'(2t + \omega_{\mathcal{H}}) + t$ vertices. □

Now, let us prove the claimed bound on $|Y_0|$ by making use of the assumed bounds $\omega_{\mathcal{H}}$ and $f_E(\cdot)$ imposed on graphs of the witness class $\mathcal{H}$.

Lemma 10. *The number of bags marked by Algorithm 1 to obtain $Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ is at most $2f_E(s(k))$, and therefore $|Y_0| \leqslant x_k = s(k) + 2f_E(s(k)) \cdot t$.*

Proof. For each bag marked in the "Large-subgraph marking step" of the algorithm, a connected subgraph C of $G - X$ with $|N_X(C)| \geqslant \omega_{\mathcal{H}}$ is found. Suppose that the algorithm finds p such connected subgraphs $C_1, \dots, C_p$. Then the number of marked bags is at most $2p$, since the LCA marking step can at most double the number of marked bags.

By the design of Algorithm 1, the connected subgraphs C_i are pairwise vertex-disjoint and $|C_i| \leqslant \rho'(2t + \omega_{\mathcal{H}}) + t$, for all $1 \leqslant i \leqslant p$, cf. Lemma 9. Define $\mathcal{P}$ to be a largest collection of paths such that the following conditions hold. For each path $P \in \mathcal{P}$:

- the endpoints of P are both in X;
- the inner vertices of P are all in a single subgraph C_i, for some $1 \leqslant i \leqslant p$; and
- for all $P' \in \mathcal{P}$ with $P' \neq P$, the endpoints of P and P' are not identical and their inner vertices are in different subgraphs C_i and C_j.

First, we show that any largest collection $\mathcal{P}$ of paths satisfying the above conditions is such that $|\mathcal{P}| = p$, that is, such a collection has one path per subgraph in $\{C_1, \dots, C_p\}$. Assume that $\mathcal{P}$ is a largest collection of paths satisfying the conditions stated above and consider the graph $H = G|_{\mathcal{P}}[X]$ induced by the vertex set X in the graph $G|_{\mathcal{P}}$ obtained by constricting the paths in $\mathcal{P}$. By assumption, $H \in \mathcal{H}$ as $\mathcal{G}$ d-constricts into $\mathcal{H}$ and $\mathcal{H}$ is closed under taking subgraphs. The constant d is given by

$$d = \max_{P \in \mathcal{P}} |P| \leqslant \max_{1 \leqslant i \leqslant p} |C_i| \leqslant \rho'(2t + \omega_{\mathcal{H}}) + t.$$

Suppose that $|\mathcal{P}| < p$, i.e., there exists some C_i with $1 \leqslant i \leqslant p$ such that no path of $\mathcal{P}$ uses vertices of C_i. Consider the neighborhood $Z = N_X^G(C_i)$ of C_i in X. As we chose the threshold of the marking algorithm to ensure that $|Z| \geqslant \omega_{\mathcal{H}}$, it follows that Z cannot induce a clique in H. But then there exist vertices $u, v \in Z$ with $uv \notin E(H)$ and we could add a uv-path whose inner vertices are in C_i to $\mathcal{P}$ without conflicting with any of the above constraints (including the bound on d), which contradicts our assumption that $\mathcal{P}$ is of largest size. We therefore conclude that $|\mathcal{P}| = p$.

Since there is a bijection from the collection of subgraphs $\{C_1, \dots, C_p\}$ and the paths of $\mathcal{P}$, we may bound p by the number of edges in H, which is at most $f_E(|H|)$. But $|H| = |X| = s(k)$ and we thus obtain the bound $p \leqslant f_E(s(k))$ on the *number* of large-degree subgraphs found by Algorithm 1. Therefore the number of marked bags is $|\mathcal{M}| \leqslant 2f_E(s(k))$. As every marked bag adds at most t vertices to Y_0, we obtain the claimed bound

$$|Y_0| = |X| + t \cdot |\mathcal{M}| \leqslant s(k) + 2t \cdot f_E(s(k)) = x_k. \qquad \square$$

We will now use this bound on the size of Y_0 to bound the sizes of the clusters $Y_1 \uplus \cdots \uplus Y_\ell$ of $G - Y_0$. The important properties used are that the instance (G, k) is reduced, that each Y_i has a small neighborhood in Y_0 and hence has small size, and that the witness graph obtained from G via constrictions has a bounded number of cliques, given by the function $f_{\#\omega}(\cdot)$.

Lemma 11. *The number of vertices in $\bigcup_{1 \leqslant i \leqslant \ell} Y_i$ is bounded by $(f_{\#\omega}(|Y_0|) + f_E(|Y_0|) + |Y_0| + 1) \cdot \rho'(2t + \omega_{\mathcal{H}})$.*

Proof. The clusters $Y_1, \dots, Y_\ell$ contain connected components of $G - Y_0$ and have the property that for each $1 \leqslant i \leqslant \ell$, $|N_{Y_0}^G(Y_i)| \leqslant 2t + \omega_{\mathcal{H}}$. We proceed analogously to the proof of Lemma 10. Let $\mathcal{P}$ be a maximum collection of paths P such that the endvertices of P are in Y_0 and all its inner vertices are in some cluster Y_i. Moreover for all paths $P_1, P_2 \in \mathcal{P}$, with $P_1 \neq P_2$, each path has a distinct set of endvertices and a distinct component for their inner vertices. Note that some clusters might have no or only one neighbor in Y_0, those cannot be used by any of the paths in $\mathcal{P}$.

Consider the graph $H = G|_{\mathcal{P}}[Y_0]$ induced by Y_0 in the graph obtained from G by constricting the paths in $\mathcal{P}$. Note that for each cluster Y_i whose vertices do not participate in any path of $\mathcal{P}$ it holds that $Z_i = N^G_{Y_0}(Y_i)$ induces a clique in H as otherwise we could augment $\mathcal{P}$ by another path. The neighborhoods of clusters not participating in any path of $\mathcal{P}$ therefore can be upperbounded by the number of cliques in H. The neighborhoods of clusters that *do* participate in paths of $\mathcal{P}$ in turn are upperbounded by $|\mathcal{P}|$, which in turns is at most $|E(H)|$.

Using the bounds $f_{\#\omega}, f_E$ for $\mathcal{H}$, we deduce that the collection of neighborhoods $\{Z_1, \dots, Z_\ell\}$, where $Z_i = N^G_{Y_0}(Y_i)$ for $1 \leqslant i \leqslant \ell$, contains at most $f_{\#\omega}(|H|) + f_E(|H|) + |H| + 1$ distinct sets where the sum $|H| + 1$ takes care of clusters that have zero to one neighbor in Y_0. Thus

$$\ell \leqslant f_{\#\omega}(|H|) + f_E(|H|) + |H| + 1 = f_{\#\omega}(|Y_0|) + f_E(|Y_0|) + |Y_0| + 1,$$

where we used the fact that $|H| = |Y_0|$. Since $Y_1, \dots, Y_\ell$ are clusters w.r.t. Y_0, we obtain ℓ restricted $(2t + \omega_{\mathcal{H}})$-protrusions in G (adding the respective neighborhood in Y_0 to each cluster yields the corresponding $(2t+\omega_{\mathcal{H}})$-protrusion). Thus the sets $Y_1, \dots, Y_\ell$ contain in total at most

$$|\bigcup_{1\leqslant i\leqslant \ell} Y_i\,| \leqslant (f_{\#\omega}(|Y_0|) + f_E(|Y_0|) + |Y_0| + 1) \cdot \rho'(2t + \omega_{\mathcal{H}})$$

vertices. □

We can now easily prove Theorem 1.

Proof of Theorem 1. By Lemma 10 we know that $|Y_0| = x_k$. Together with Lemma 11 we can bound the total number of vertices in a reduced instance by

$$\begin{aligned} |V(G)| &= |Y_0| + |Y_1| + \ldots + |Y_\ell| \\ &\leqslant x_k + (f_{\#\omega}(x_k) + f_E(x_k) + x_k + 1)\rho'(2t + \omega_{\mathcal{H}}), \end{aligned}$$

again using the shorthand $x_k = s(k) + 2f_E(s(k)) \cdot t$. □

We now show how to apply Theorem 1 to obtain kernels. Let $\mathcal{G}_H$ be the class of graphs that exclude some fixed graph H as a topological minor. Observe that $\mathcal{G}_H$ is closed under taking topological minors, and is therefore closed under taking d-constrictions for any $d \geqslant 2$.

In order to obtain $f_E, f_{\#\omega}$, and $\omega_{\mathcal{G}_H}$ we use the fact that H-topological-minor-free graphs are *ε-degenerate.* That is, there exists a constant ε (that depends only on H) such that every subgraph of $G \in \mathcal{G}_H$ contains a vertex of degree at most ε. The following are well-known properties of degenerate graphs.

Proposition 2 (Bollobás and Thomason [13], Komlós and Szemerédi [58])**.** *There is a constant $\beta \leqslant 10$ such that, for $r > 2$, every graph with no K_r-topological-minor has average degree at most βr^2.*

As an immediate consequence, any graph with average degree larger than βr^2 contains *every* r-vertex graph as a topological minor. If a graph G excludes H as a topological minor, then G clearly excludes $K_{|H|}$ as a topological minor. What is also true is that the *total* number of cliques (not necessarily maximal) in G is $O(|G|)$.

Proposition 3 (Fomin, Oum, and Thilikos [44])**.** *There is a constant $\tau < 4.51$ such that, for $r > 2$, every n-vertex graph with no K_r-topological-minor has at most $2^{\tau r \log r} n$ cliques.*

Henceforth, let $r := |H|$ denote the size of the forbidden topological minor. The following is a slightly generalized version of our first main theorem.

Theorem 2. *Fix a graph H and let $\mathcal{G}_H$ be the class of H-topological-minor-free graphs. Let Π be a parameterized graph problem that has finite integer index and is $(s_{\Pi,\mathcal{G}_H}, t_{\Pi,\mathcal{G}_H})$-treewidth-bounding on the class $\mathcal{G}_H$. Then Π admits a kernel of size $O(s_{\Pi,\mathcal{G}_H}(k))$.*

Proof. We use Theorem 1 with the functions $f_E(n) = \frac{1}{2}\beta r^2 n, f_{\#\omega}(n) = 2^{\tau r \log r} n$ obtained from Propositions 2 and 3. Observe that an H-topological-minor-free graph cannot contain a clique of size r, thus $\omega_{\mathcal{G}_H} \leqslant r$. The kernel size is then bounded by

$$s_{\Pi,\mathcal{G}_H}(k) \cdot \Big(1 + \beta r^2 t\Big) \Big(1 + (2^{\tau r \log r} + \frac{1}{2}\beta r^2 + 1)\rho'(2t + r)\Big) + \rho'(2t + r)$$

$$s_{\Pi,\mathcal{G}_H}(k) \cdot (1 + \beta r^2 t + (2^{\tau r \log r}(1 + \beta r^2 t) + \beta r^2 t) \cdot \rho'(2t + r)) + \rho'(2t + r),$$

where we omitted the subscript of $t_{\Pi,\mathcal{G}_H}$ for the sake of readability. □

Theorem I is now just a consequence of the special case for which the treewidth-bound is linear. Note that the class of graphs with bounded degree is a subset of those that exclude a fixed topological minor, thus the above result translates directly to this class.

4.2 Problems affected by our result

We present concrete problems that satisfy the prerequisites of Theorem I.

Corollary 1. *Fix a graph H. The following problems are linearly treewidth-bounding and have finite integer index and linear treewidth-bound on the class of H-topological-minor-free graphs and hence possess a linear kernel on this graph class:* Vertex Cover[8]*;* Cluster Vertex Deletion[8]*;* Feedback Vertex Set*;* Chordal Vertex Deletion[9]*;* Interval *and* Proper Interval Vertex Deletion*;* Cograph Vertex Deletion*;* Edge Dominating Set.

In particular, Corollary 1 also implies that Chordal Vertex Deletion and Interval Vertex Deletion can be decided on H-topological-minor-free graphs in time $O(c^k \cdot poly(n))$ for some constant c. (This follows because one can first obtain a linear kernel and then use brute-force to solve the kernelized instance.) On general graphs (complicated) single-exponential algorithms for Interval Vertex Deletion have appeared only very recently [17, 18, 75] (until now, this problem was not even known to be FPT), whereas only an $O(f(k) \cdot poly(n))$ algorithm is known for Chordal Vertex Deletion, where $f(k)$ is not even specified [66].

Corollary 2. Chordal Vertex Deletion *and* Interval Vertex Deletion *are solvable in single-exponential time on H-topological-minor-free graphs.*

A natural extension of the (vertex deletion) problems in Corollary 1 is to seek a solution that induces a *connected* graph. The connected versions of problems are typically more difficult both in terms of proving fixed-parameter tractability and establishing polynomial kernels. For instance, Vertex Cover admits a $2k$-vertex kernel but Connected Vertex Cover has no polynomial kernel unless NP $\subseteq$ co-NP/*poly* [30]. However on H-topological-minor-free graphs, Connected Vertex Cover (and a couple of others) admit a linear kernel.

[8]Listed for completeness; these problems have a kernel with a linear number of vertices on general graphs.

[9]Note that Chordal Vertex Deletion is indeed linearly treewidth-bounding on H-topological-minor-free graphs, since an H-topological-minor-free chordal graph has bounded clique size, and hence bounded treewidth as well.

Corollary 3. Connected Vertex Cover, Connected Cograph Vertex Deletion, *and* Connected Cluster Vertex Deletion *have linear kernels in graphs excluding a fixed topological minor.*

Another property of H-topological-minor-free graphs is that the well-known graph width measures treewidth ($\mathbf{tw}$), rankwidth ($\mathbf{rw}$), and cliquewidth ($\mathbf{cw}$), are all within a constant multiplicative factor of one another.

Proposition 4 (Fomin, Oum, and Thilikos [44]). *There is a constant τ such that for every $r > 2$, if G excludes K_r as a topological minor, then*

$$\mathbf{rw}(G) \leqslant \mathbf{cw}(G) < 2 \cdot 2^{\tau r \log r}\, \mathbf{rw}(G)$$
$$\mathbf{rw}(G) \leqslant \mathbf{tw}(G) + 1 < \frac{3}{4}(r^2 + 4r - 5)2^{\tau r \log r}\, \mathbf{rw}(G).$$

An interesting vertex-deletion problem related to graph width measures is Width-b Vertex Deletion [56]: given a graph G and an integer k, do there exist at most k vertices whose deletion results in a graph with width at most b? From Definition 10 (see Section 2), it follows that if the width measure is treewidth, then this problem is treewidth-bounding. By Proposition 4, this also holds if the width measure is either rankwidth or cliquewidth. The fact that this problem has finite integer index follows from the sufficiency condition known as *strong monotonicity* in [8]. Since branchwidth differs only by a constant factor from treewidth in general graphs [79], this gives us the following.

Corollary 4. *The* Width-b Vertex Deletion *problem has a linear kernel on H-topological-minor-free graphs, where the width measure is either treewidth, cliquewidth, branchwidth, or rankwidth.*

4.3 A comparison with earlier results

We briefly compare the structural constraints imposed in Theorem I with those imposed in the results on linear kernels on graphs of bounded genus [8] and H-minor-free graphs [42]. In particular, we discuss how restrictive is the condition of being treewidth-bounding. A graphical summary of the various notions of sparseness and the associated structural constraints used to obtain results on linear kernels is depicted in Figure 3.

The theorem that guarantees linear kernels on graphs of bounded genus in [8] imposes a condition called *quasi-compactness*. The notion of quasi-compactness is similar to that of treewidth-bounding: Yes-instances (G, k) satisfy the condition that there exists a vertex set $X \subseteq V(G)$ of "small" size whose deletion yields a graph of bounded treewidth. Formally, a problem Π is called *quasi-compact* if there exists an integer r such that for every $(G, k) \in \Pi$, there is an embedding of G onto a surface of Euler-genus at most g and a set $X \subseteq V(G)$ such that $|X| \leqslant r \cdot k$ and $\mathbf{tw}(G - R^r_G(X)) \leqslant r$. Here $R^r_G(X)$ denotes the set of vertices of G at radial distance at most r from X. It is easy to see that the property of being treewidth-bounding is stronger than quasi-compactness in the sense that if a problem is treewidth-bounding and the graphs are embeddable on a surface of genus g, then the problem is also quasi-compact, but not the other way around. The fact that we use a stronger structural condition is expected, since our result proves a linear kernel on a much larger graph class.

More interesting are the conditions imposed for linear kernels on H-minor-free graphs [42]. The problems here are required to be *bidimensional* and satisfy a so-called *separation property*. Roughly speaking, a problem is bidimensional if the solution size on a $k \times k$-grid is $\Omega(k^2)$ and the solution

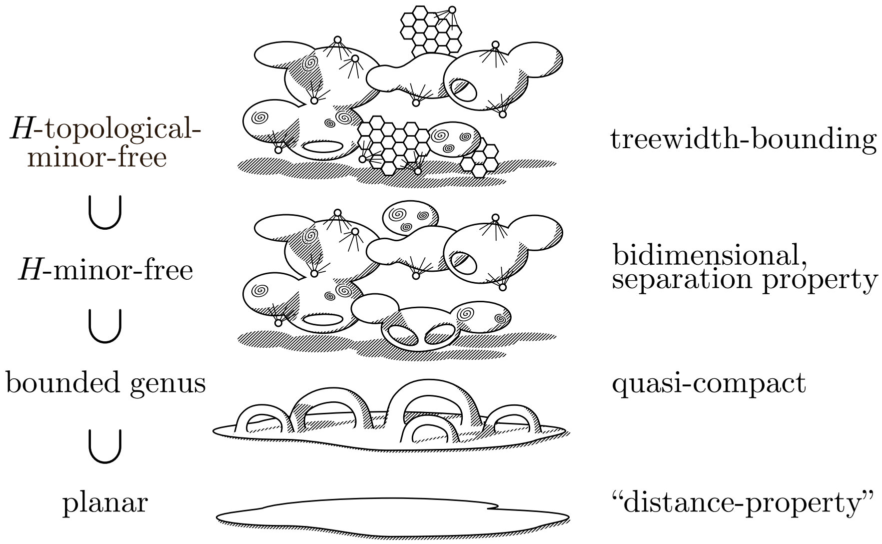

Figure 3: Kernelization results for problems with finite integer index on sparse graph classes with their corresponding additional condition.

size does not decrease by deleting/contracting edges. The notion of the separation property is essentially the following. A problem has the separation property, if for any graph G and any vertex subset $X \subseteq V(G)$, the optimum solution of G projected on any subgraph G' of $G - X$ differs from the optimum for G' by at most $|X|$ (cf. [42] for details.) At first glance, these conditions seem to have nothing to do with the property of being treewidth-bounding. However in the same paper [42, Lemma 3.2], the authors show that if a problem on H-minor-graphs is bidimensional and has the separation property then it is also (ck, t)-treewidth-bounding for some constants c, t that depend on the graph H excluded as a minor. Using this fact, the main result of [42] (namely, that bidimensional problems with FII and the separation property have linear kernels on H-minor-free graphs) can be reproved as an easy corollary of Theorem 1.

This discussion shows that in the results on linear kernels on sparse graph classes that we know so far, the treewidth-bounding condition has appeared in some form or the other. In the light of this we feel that this is the key condition for proving linear kernels on sparse graph classes.

4.4 The limits of our approach

It is interesting to know for which notions of sparseness (beyond H-topological-minor-free graphs) we can use our technique to obtain polynomial kernels. We show that our technique *fails* for the following notion of sparseness: graph classes that locally exclude a minor [25]. The notion of locally excluding a minor was introduced by Dawar *et al.* [25] and graphs that locally exclude a minor include bounded-genus graphs but are incomparable with H-minor-free graphs [71]. However we also show that there exist (restricted) graph classes that locally exclude a minor where it is still possible to obtain a polynomial kernel using our technique.

Definition 12 (Locally excluding a minor [25])**.** A class $\mathcal{G}$ of graphs *locally excludes a minor* if for every $r \in \mathbb{N}$ there is a graph H_r such that the r-neighborhood of a vertex of any graph of $\mathcal{G}$ excludes H_r as a minor.

Therefore if $\mathcal{G}$ locally excludes a minor then the 1-neighborhood of a vertex in any graph of $\mathcal{G}$ does not contain H_1 as a minor, and hence as a subgraph. In particular, the neighborhood of no vertex contains a clique on $h_1 := |H_1|$ vertices as a subgraph, meaning that the clique number (that is, the maximum size of a clique) of such graphs is bounded from above by h_1. The total number of cliques in any graph of $\mathcal{G}$ is then bounded by $h_1 n^{h_1}$, and the number of edges can be trivially bounded by n^2. We now have almost all the prerequisites for applying Theorem 1. However the class $\mathcal{G}$ is not closed under taking d-constrictions. Taking a d-constriction in a graph $G \in \mathcal{G}$ can increase the clique number of the constricted graph. This seems to be a bottleneck in applying Theorem 1. However if we assume that the size of the locally forbidden minors $\{H_r\}_{r\in\mathbb{N}}$ grows very slowly, then we can still obtain a polynomial kernel.

Definition 13. Given $g\colon \mathbb{N} \to \mathbb{N}$, we say that a graph class $\mathcal{G}$ *locally excludes minors according to* g if there exists a constant $n_0 \in \mathbb{N}$, such that for all $r \geqslant n_0$, the $g(r)$-neighborhood of a vertex in any graph of $\mathcal{G}$ does not contain K_r as a minor.

Lemma 12. *Let $\mathcal{G}$ be a graph class that locally excludes a minor according to $g\colon \mathbb{N} \to \mathbb{N}$ and let n_0 be the constant as in the above definition. Then for any $r \geqslant n_0$, the class $\mathcal{G}$ $g(r)$-constricts into a graph class $\mathcal{H}$ that excludes K_r as a subgraph.*

Proof. Assume the contrary. Let $G \in \mathcal{G}$ and suppose that for some $r \geqslant 2$ the graph H obtained by a $g(r)$-constriction of G contains K_r as a subgraph. Pick any vertex v in this subgraph of H. The $g(r)$-neighborhood of v in G must contain K_r as a minor, a contradiction. □

Note that in the following, we assume that the problem is treewidth-bounding on general graphs.

Corollary 5. *Let Π be a parameterized graph problem with finite integer index that is $(s(k), t_\Pi)$-treewidth-bounding. Let $\mathcal{G}$ be a graph class locally excluding a minor according to a function $g\colon \mathbb{N} \to \mathbb{N}$ such that for all $r \geqslant n_0$, $g(r) \geqslant \rho'(2t_\Pi + r) + 1$. Then there exists a constant r_0 such that Π admits kernels of size $O(s(k)^{r_0})$ on $\mathcal{G}$.*

Proof. By Lemma 12, taking a $g(r)$-constriction results in a graph class $\mathcal{H}$ that excludes K_r as a subgraph, for large enough r. Fixing $r = n_0$, where n_0 is the constant in Definition 13, we apply Theorem 1 with the trivial functions $f_E(n) = n^2$, $f_{\#\omega}(n) = r \cdot n^r$ and $\omega_{\mathcal{H}} = r$. By Lemma 12, we have that $\omega_{\mathcal{G}_H} \leqslant r$. The kernel size is then bounded by

$$s(k)+2ts(k)^2+\Big((s(k) + 2t \cdot s(k)^2)^r + (s(k) + 2ts(k)^2)^2 + s(k) + 2ts(k)^2 + 1\Big)\,\rho'(2t+r) \in O(s(k)^{2r}),$$

where we omitted the subscript of t_Π for the sake of readability. With $r_0 = 2r = 2n_0$, the bound in the statement of the corollary follows. □

We do not know how quickly the function $\rho'(\cdot)$ grows but intuition from automata theory seems to suggest that this has at least superexponential growth. As such, the graph class for which the polynomial kernel result holds (Corollary 5) is pretty restricted. However this does suggest a limit to which our approach can be pushed as well as some intuition as to why our result is not easily extendable to graph classes locally excluding a minor. We note that graph classes of bounded expansion present the same problem.

4.5 An illustrative example: Edge Dominating Set

In this section we show how Theorem I can actually be used to obtain a simple explicit kernel for the Edge Dominating Set problem on H-topological-minor-free graphs. This is made possible by the fact that we can find in polynomial time a small enough treewidth-modulator *and* replace the generic protrusion reduction rule by a handcrafted specific reduction rule.

Let us first recall the problem at hand. We say that an edge e is *dominated* by a set of edges D if either $e \in D$ or e is incident with at least one edge in D. The problem Edge Dominating Set asks, given a graph G and an integer k, whether there is an edge dominating set $D \subseteq E(G)$ of size at most k, i.e., an edge set which dominates every edge of G. The canonical parameterization of this problem is by the integer k, i.e., the size of solution set.

There is a simple 2-approximation algorithm for Edge Dominating Set [85]. Given an instance (G, k), where G is H-topological-minor-free, let D be an edge dominating set of G, given by the 2-approximation. We can assume that $|D| \leqslant 2k$ since otherwise we can correctly declare (G, k) as a No-instance. Take $X := \{v \in V(G) \mid v \text{ is incident to some edge in } D\}$ as the treewidth-modulator: note that $|X| \leqslant 4k$ and that $G - X$ is of treewidth at most 0, i.e., an independent set. One can easily verify that that the bag marking Algorithm 1 of Section 3 would mark exactly those vertices of $G - X$ whose neighborhood in X has size at least $r := |H|$. By applying the edge-bound of Proposition 2 to Lemma 10 we get that $|V(\mathcal{M})| \leqslant \beta r^2 \cdot 8k$.

Take $Y_0 := X \cup V(\mathcal{M})$ and let $\mathcal{P} := Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ be a partition of $V(G)$, where again Y_i, $1 \leqslant i \leqslant \ell$, is now a cluster w.r.t. Y_0, i.e., the vertices in a single Y_i share the same neighborhood in X and the Y_i are of maximal size under this condition. We have one reduction rule, which can be construed as an concrete instantiation of generic the protrusion replacement rule. We would like to stress that this reduction rule relies on the fact that we already have a protrusion decomposition of G, given by Algorithm 1.

> **Twin elimination rule**: If $|Y_i| > |N_{Y_0}(Y_i)|$ for some $i \neq 0$, let G' be the instance obtained by keeping $|N_{Y_0}(Y_i)|$ many vertices of Y_i and removing the rest of Y_i. Take $k' := k$.

Lemma 13. *The twin elimination rule is safe.*

Proof. Let G_i be the graph induced by the vertex set $N_{Y_0}(Y_i) \cup Y_i$ and let E_i be its edge set (as $N_{Y_0}(Y_i) = N(Y_i)$, we shall omit the subscript Y_0). For a vertex $v \in V(G)$, we define the set $E(v)$ as the set of edges incident with v. The notations G'_i, E'_i, Y'_i, and $E'(v)$ are defined analogously for the graph G' obtained after the application of twin elimination rule. We say that a vertex $v \in V(G)$ is *covered* by an edge set D if v is incident with an edge of D.

To see the forward direction, suppose that (G, k) is a Yes-instance and let D be an edge dominating set of size at most k. Without loss of generality, we can assume that $|D \cap E_i| \leqslant |N(Y_i)|$. Indeed, it can be easily checked that the edge set $(D \setminus E_i) \cup E(u)$, for an arbitrarily chosen $u \in Y_i$, is an edge dominating set. Hence at most $|N(Y_i)|$ vertices out of Y_i are covered by D, and thus we can apply twin elimination rule so as to delete only those vertices which are not incident with D. It just remains to observe that D is an edge dominating set of G'.

For the opposite direction, let D' be an edge dominating set for G' of size at most k. We first argue that $N(Y'_i)$ is covered by D' without loss of generality. Indeed, suppose $v \in N(Y'_i)$ is not covered by D'. In order for an edge $e = uv \in E'(v) \cap E'_i$ to be dominated by D', at least one edge in $E'(u)$ should be contained in D. Since the sets $\{E'(u) : u \in Y'_i\}$ are mutually disjoint, it follows that $|D' \cap E'_i| \geqslant |Y'_i|$. Now take an alternative edge set $D'' := (D' \setminus E'_i) \cup E'(u)$ for an arbitrary vertex $u \in Y'_i$. It is not difficult to see that D'' is an edge dominating set for G'. Moreover, we have $|D''| \leqslant |D'| \leqslant k$ as $|D' \cap E'_i| \geqslant |Y'_i| = |E'(u)| = |N(Y'_i)|$. Hence D'' is also an edge dominating set of size at most k. Assuming that $N(Y'_i)$ is covered by D', it is easy to see that D' dominates E_i and thus D' is an edge dominating set of G. This complete the proof. $\square$

Back to the partition $\mathcal{P}$, we can apply the twin elimination rule in time $O(n)$ and ensure that $|Y_i| \leqslant r-1$ for $1 \leqslant i \leqslant \ell$. The bound on ℓ is proved in Lemma 11 and taken together with the edge- and clique-bounds from Proposition 2 and 3, respectively, we obtain

$$\begin{aligned} \ell &\leqslant 2^{\tau r \log r}((2\beta r^2+1)4k) + 2\beta r^2((2\beta r^2+1)4k) + (2\beta r^2+1)4k + 1 \\ &= 4k(\beta r^2 2^{\tau r \log r + 1} + 4\beta^2 r^4 + 4\beta r^2 + 2^{\tau r \log r} + 1) + 1 \end{aligned}$$

and thus we get the overall bound

$$\begin{aligned} |G| &\leqslant |Y_0| + |Y_1| + \ldots + |Y_\ell| \\ &\leqslant 4k(\beta r^2+1) + \Big(4k(\beta r^2 2^{\tau r \log r+1} + 4\beta^2 r^4 + 4\beta r^2 + 2^{\tau r \log r} + 1) + 1\Big)(r-1) \\ &\leqslant 4k((\beta r^2 2^{\tau r \log r+1} + 4\beta^2 r^4 + 4\beta r^2 + 2^{\tau r \log r} + 1)(r-1) + 2\beta r^2 + 1) + (r-1) \\ &< k\Big((80r^2 20.8^{r \log r+1} + 6400r^4 + 320r^2 + 4 \cdot 28.8^{r \log r} + 4)(r-1) + 160r^2 + 4\Big) + r \end{aligned}$$

on the size of G. We remark that this upper bound can be easily made explicit once H is fixed. Again, we can get better constants on H-minor-free graphs, just by replacing constants βr^2 and $2^{\tau r \log r}$ with $\alpha(r\sqrt{\log r})$ and $2^{\mu r \log\log r}$, respectively. Finally, note that the whole procedure can be carried out in linear time.

5 Single-exponential algorithm for Planar-$\mathcal{F}$-Deletion

This section is devoted to the single-exponential algorithm for the Planar-$\mathcal{F}$-Deletion problem. Let henceforth H_p be some fixed (connected or disconnected) arbitrary planar graph in the family $\mathcal{F}$, and let $r := |H_p|$. First of all, using iterative compression, we reduce the problem to obtaining a single-exponential algorithm for the Disjoint Planar-$\mathcal{F}$-Deletion problem, which is defined as follows:

Disjoint Planar-$\mathcal{F}$-Deletion
Input: A graph G and a subset of vertices $X \subseteq V(G)$ such that $G - X$ is H-minor-free for every $H \in \mathcal{F}$.
Parameter: The integer $k = |X|$.
Objective: Compute a set $\tilde{X} \subseteq V(G)$ disjoint from X such that $|\tilde{X}| < |X|$ and $G - \tilde{X}$ is H-minor-free for every $H \in \mathcal{F}$, if such a set exists.

The input set X is called the *initial solution* and the set $\tilde{X}$ the *alternative solution*. Let $t_{\mathcal{F}}$ be a constant (depending on the family $\mathcal{F}$) such that $\mathbf{tw}(G-X) \leqslant t_{\mathcal{F}} - 1$ (note that such a constant exists by Robertson and Seymour [77], and can be calculated by using the results of Fellows and Langston [34]).

The following lemma relies on the fact that being $\mathcal{F}$-minor-free is a hereditary property with respect to induced subgraphs. For a proof, see for instance [20, 53, 56, 65].

Lemma 14. *If the parameterized* Disjoint Planar-$\mathcal{F}$-Deletion *problem can be solved in time* $c^k \cdot p(n)$*, where* c *is a constant and* $p(n)$ *is a polynomial in* n*, then the parameterized* Planar-$\mathcal{F}$-Deletion *problem can be solved in time* $(c+1)^k \cdot p(n) \cdot n$*.*

Let us provide a brief sketch of our algorithm to solve Disjoint Planar-$\mathcal{F}$-Deletion. We start by computing a protrusion decomposition using Algorithm 1 with input (G, X, r). But it turns out that the set Y_0 output by Algorithm 1 does not define a *linear* protrusion decomposition of G, which is crucial for our purposes (in fact, it can be only proved that Y_0 defines a *quadratic* protrusion decomposition of G). To circumvent this problem, our strategy is to first use Algorithm 1 to identify a set Y_0 of $O(k)$ vertices of G, and then guess the intersection I of the alternative solution $\tilde{X}$ with the set Y_0. We prove that if the input is a Yes-instance of Disjoint Planar-$\mathcal{F}$-Deletion, then $V(\mathcal{M})$ contains a subset I such that the connected components of $G - V(\mathcal{M})$ can be clustered together with respect to their neighborhood in $Y_0 \setminus I$ to form an $(O(k - |I|), 2t_{\mathcal{F}} + r)$-protrusion decomposition $\mathcal{P}$ of the graph $G - I$. As a result, we obtain Proposition 5, which is fundamental in order to prove Theorem II.

Proposition 5 (**Linear protrusion decomposition**)**.** *Let* (G, X, k) *be a* Yes*-instance of the parameterized* Disjoint Planar-$\mathcal{F}$-Deletion *problem. There exists a* $2^{O(k)} \cdot n$*-time algorithm that identifies a set* $I \subseteq V(G)$ *of size at most* k *and a* $(O(k), 2t_{\mathcal{F}} + r)$*-protrusion decomposition* $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ *of* $G - I$ *such that:*

1. $X \subseteq Y_0$*;*
2. *there exists a set* $X' \subseteq V(G) \setminus Y_0$ *of size at most* $k - |I|$ *such that* $G - \tilde{X}$*, with* $\tilde{X} = X' \cup I$*, is* H*-minor-free for every graph* $H \in \mathcal{F}$*.*

At this stage of the algorithm, we can assume that a subset I of the alternative solution $\tilde{X}$ has been identified, and it remains to solve the instance $(G - I, X, k - |I|)$ of the Disjoint Planar-$\mathcal{F}$-Deletion problem, which comes equipped with a linear protrusion decomposition $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$. In order to solve this problem, we prove the following proposition:

Proposition 6. *Let* (G, Y_0, k) *be an instance of* Disjoint Planar-$\mathcal{F}$-Deletion *and let* $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ *be an* (α, β)*-protrusion decomposition of* G*, for some constant* β*. There exists an* $2^{O(\ell)} \cdot n$*-time algorithm that computes a solution* $\tilde{X} \subseteq V(G) \setminus Y_0$ *of size at most* k *if it exists, or correctly decides that there is no such solution.*

The key observation in the proof of Proposition 6 is that for every restricted protrusion Y_i, there is a *finite* number of representatives such that any partial solution lying on Y_i can be replaced with one of these while preserving the feasibility of the solution. This follows from the *finite index* of MSO-definable properties (see, e.g., [12]). Then, to solve the problem in single-exponential time we can just use brute-force in the union of these representatives, which has overall size $O(k)$.

Organization of the section. In Subsection 5.1 we analyze Algorithm 1 when the input graph is a Yes-instance of Disjoint Planar-$\mathcal{F}$-Deletion. The branching step guessing the intersection of the alternative solution $\tilde{X}$ with $V(\mathcal{M})$ is described in Subsection 5.2, concluding the proof of Proposition 5. Subsection 5.3 gives a proof of Proposition 6, and finally Subsection 5.4 proves Theorem II.

5.1 Analysis of the bag marking algorithm

We first need two results concerning graphs which exclude a clique as a minor. The following lemma states that graphs excluding a fixed graph as a minor have linear number of edges.

Proposition 7 (Thomason [83])**.** *There is a constant $\alpha < 0.320$ such that every n-vertex graph with no K_r-minor has at most $(\alpha r\sqrt{\log r}) \cdot n$ edges.*

Recall that a *clique* in a graph is a set of pairwise adjacent vertices. For simplicity, we assume that a single vertex and the empty graph are also cliques.

Proposition 8 (Fomin, Oum, and Thilikos [44])**.** *There is a constant $\mu < 11.355$ such that, for $r > 2$, every n-vertex graph with no K_r-minor has at most $2^{\mu r \log\log r} \cdot n$ cliques.*

For the sake of simplicity, let henceforth in this section $\alpha_r := \alpha r\sqrt{\log r}$ and $\mu_r := 2^{\mu r \log\log r}$.

Let us now analyze some properties of Algorithm 1 when the input graph is a Yes-instance of the Disjoint Planar-$\mathcal{F}$-Deletion problem. In this case, the bound on the treewidth of $G - X$ is $t_{\mathcal{F}} - 1$. The following two lemmas show that the number of bags identified at the "Large-subgraph marking step" is linearly bounded by k. Their proofs use arguments similar to those used in the proof of Theorem 1, but we provide the full proofs here for completeness.

Lemma 15. *Let (G, X, k) be a* Yes*-instance of the* Disjoint Planar-$\mathcal{F}$-Deletion *problem. If $C_1, \ldots, C_\ell$ is a collection of connected pairwise disjoint subsets of $V(G) \setminus X$ such that for all $1 \leqslant i \leqslant \ell$, $|N_X(C_i)| \geqslant r$, then $\ell \leqslant (1 + \alpha_r) \cdot k$.*

Proof. Let $X' \subseteq V(G) \setminus X$ be a solution for (G, X, k), and observe that $\ell' \leqslant k$ of the sets $C_1, \ldots, C_\ell$ contain vertices of X'. Consider the sets $C_{\ell'+1}, \ldots, C_\ell$ which are disjoint with X', and observe that $G[X \cup (\bigcup_{\ell' < j \leqslant \ell} C_j)]$ is an H-minor-free graph. We proceed to construct a family of graphs $\{G_i\}_{\ell' \leqslant i \leqslant \ell}$, with $V(G_i) = X$ for all $\ell' \leqslant i \leqslant \ell$, and such that G_i is a minor of $G[X \cup (\bigcup_{\ell' < j \leqslant i} C_j)]$, in the following way. We start with $E(G_{\ell'}) = E[G(X)]$, and suppose inductively that the graph G_{i-1} has been successfully constructed. Since by assumption G_{i-1} is a minor of $G[X \cup (\bigcup_{\ell' < j \leqslant i-1} C_j)]$, which in turn is a minor of $G[X \cup (\bigcup_{\ell' < j \leqslant \ell} C_j)]$, it follows that G_{i-1} is H-minor-free, and therefore it cannot contain a clique on r vertices. In order to construct G_i from G_{i-1}, let x_i, y_i be two vertices in X such that both x_i and y_i are neighbors in G of some vertex in C_i, and such that x_i and y_i are non-adjacent in G_{i-1}. Note that such two vertices exist, since we can assume that $r \geqslant 2$ and G_{i-1} is H-minor-free. Then G_i is constructed from G_{i-1} by adding an edge between x_i and y_i. Since C_i is connected by hypothesis, we have that G_i is indeed a minor of $G[X \cup (\bigcup_{\ell' < j \leqslant i} C_j)]$. Since G_ℓ is H-minor-free, it follows by Proposition 7 that $|E(G_\ell)| \leqslant \alpha_r \cdot |X|$ edges. Since by construction we have that $\ell - \ell' \leqslant |E(G_\ell)|$, we conclude that $\ell = \ell' + (\ell - \ell') \leqslant k + \alpha_r \cdot k = (1 + \alpha_r) \cdot k$, as we wanted to prove. □

Lemma 16. *If (G, X, k) is a* Yes*-instance of the* Disjoint Planar-$\mathcal{F}$-Deletion *problem, then the set $Y_0 = V(\mathcal{M}) \cup X$ of vertices returned by Algorithm 1 has size at most $k + 2t_{\mathcal{F}} \cdot (1 + \alpha_r) \cdot k$.*

Proof. As $|X| = k$ and as the algorithm marks bags of an optimal forest-decomposition of $G - X$, which is a graph of treewidth at most $t_{\mathcal{F}}$, in order to prove the lemma it is enough to prove that the number of marked bags is at most $2 \cdot (1 + \alpha_r) \cdot k$. It is an easy observation to see that the set of connected components C_B identified at the Large-subgraph marking step contains pairwise vertex disjoint subset of vertices, each inducing a connected subgraph of $G - X$ with at least r neighbors in

X. It follows by Lemma 15, that the number of bags marked at the Large-subgraph marking step is at most $(1+\alpha_r)\cdot k$. To conclude it suffices to observe that the number of bags identified at the LCA marking step cannot exceed the number of bags marked at the Large-subgraph marking step. $\square$

5.2 Branching step and linear protrusion decomposition

At this stage of the algorithm, we have identified a set $Y_0 = X \cup V(\mathcal{M})$ of $O(k)$ vertices such that, by Proposition 1, every connected component of $G - Y_0$ is a restricted $(2t_{\mathcal{F}}+r)$-protrusion. We would like to note that it can be proved, using ideas similar to the proof of Lemma 17 below, that Y_0 together with the clusters of $G - Y_0$ form a *quadratic* protrusion decomposition of the input graph G. But as announced earlier, for time complexity issues we seek a *linear* protrusion decomposition. To this end, the second step of the algorithm consists in a branching to guess the intersection I of the alternative solution $\tilde{X}$ with the set of marked vertices $V(\mathcal{M})$. By Lemma 16, this step yields $2^{O(k)}$ branchings, which is compatible with the desired single-exponential time.

For each guessed set $I \subseteq Y_0$, we denote $G_I := G - I$. Recall that a cluster of $G_I - Y_0$ as a maximal collection of connected components of $G_I - Y_0$ with the same neighborhood in $Y_0 \setminus I$. We use Observation 2, a direct consequence of Lemma 7, to bound the number of clusters under the condition that G_I contains a vertex subset X' disjoint from Y_0 of size at most $k - |I|$ such that $G_I - X'$ does not contain any graph $H \in \mathcal{F}$ as a minor (and so the graph $G - \tilde{X}$, with $\tilde{X} = X' \cup I$, does not contain either any graph $H \in \mathcal{F}$ as a minor).

Observation 2. *For every cluster $\mathcal{C}$ of $G_I - Y_0$, $|N_{Y_0}(\mathcal{C})| < r + 2t_{\mathcal{F}}$.*

The proof of the following lemma has a similar flavor to those of Theorem 1 and Lemma 15.

Lemma 17. *If $(G_I, Y_0 \setminus I, k - |I|)$ is a* Yes*-instance of the* Disjoint Planar-$\mathcal{F}$-Deletion *problem, then the number of clusters of $G_I - Y_0$ is at most $(5t_{\mathcal{F}}\alpha_r\mu_r)\cdot k$.*

Proof. Let $\mathcal{C}$ be the collection of all clusters of $G_I - Y_0$. Let X' be a subset of vertices disjoint from Y_0 such that $|X'| \leqslant k - |I| = k_I$ and $G_I - X'$ is H-minor-free for every graph $H \in \mathcal{F}$. Observe that at most k_I clusters in $\mathcal{C}$ contain vertices from X'. Let $C_1, \ldots, C_\ell$ be the clusters in $\mathcal{C}$ that do not contain vertices from X'. So we have that $|\mathcal{C}| \leqslant k_I + \ell \leqslant k + \ell$. Let $G_{\mathcal{C}}$ be the subgraph of G induced by $(Y_0 \setminus I) \cup \bigcup_{i=1}^{\ell} C_i$. Observe that as $(G_I, Y_0 \setminus I, k - |I|)$ is a Yes-instance of the Disjoint Planar-$\mathcal{F}$-Deletion problem, $G_{\mathcal{C}}$ is H-minor-free for every graph $H \in \mathcal{F}$.

We greedily construct from $G_{\mathcal{C}}$ a graph $G'_{\mathcal{C}}$, with $V(G'_{\mathcal{C}}) = Y_0 \setminus I$, as follows. We start with $G'_{\mathcal{C}} = G[Y_0 \setminus I]$. As long as there is a non-used cluster $C \in \mathcal{C}$ with two non-adjacent neighbors u, v in $Y_0 \setminus I$, we add to $G'_{\mathcal{C}}$ an edge between u and v and mark C as used. The number of clusters in $\mathcal{C}$ used so far in the construction of $G'_{\mathcal{C}}$ is bounded from above by the number of edges of $G'_{\mathcal{C}}$. Observe that by construction $G'_{\mathcal{C}}$ is clearly a minor of $G_{\mathcal{C}}$. Thereby $G'_{\mathcal{C}}$ is an H-minor-free graph (for every $H \in \mathcal{F}$) on at most $k + 2t_{\mathcal{F}}\cdot(1+\alpha_r)\cdot k$ vertices (by Lemma 16). By Proposition 7, it follows that $|E(G'_{\mathcal{C}})| \leqslant \alpha_r \cdot (k + 2t_{\mathcal{F}}\cdot(1+\alpha_r)\cdot k)$ and so there are the same number of used clusters.

Let us now count the number of non-used clusters. Observe that the neighborhood in $Y_0 \setminus I$ of each non-used cluster induces a (possibly empty) clique in $G'_{\mathcal{C}}$ (as otherwise some further edge could have been added to $G'_{\mathcal{C}}$). As by definition distinct clusters have distinct neighborhoods in $Y_0 \setminus I$, and as $G'_{\mathcal{C}}$ is an H-minor-free graph (for every $H \in \mathcal{F}$) on at most $k + 2t_{\mathcal{F}}\cdot(1+\alpha_r)\cdot k$ vertices, Proposition 8 implies that the number of non-used clusters is at most $\mu_r \cdot (k + 2t_{\mathcal{F}}\cdot(1+\alpha_r)\cdot k)$. Summarizing, we have that $|\mathcal{C}| \leqslant k + (\alpha_r + \mu_r)\cdot(k + 2t_{\mathcal{F}}\cdot(1+\alpha_r)\cdot k) \leqslant (5t_{\mathcal{F}}\alpha_r\mu_r)\cdot k$, where in the last inequality we have used that $\mu_r \geqslant \alpha_r$ and we have assumed that $\alpha_r \geqslant 4$. $\square$

Piecing all lemmas together, we can now provide a proof of Proposition 5.

Proof of Proposition 5. By Lemmas 5 and 16 and Observation 2, we can compute in linear time a set Y_0 of $O(k)$ vertices containing X such that every cluster of $G - Y_0$ is a restricted $(2t_{\mathcal{F}} + r)$-protrusion. If (G, X, k) is a YES-instance of the DISJOINT PLANAR-$\mathcal{F}$-DELETION problem, then there exists a set $\tilde{X}$ of size less than $|X|$ and disjoint from X such that $G - \tilde{X}$ does not contain any graph $H \in \mathcal{F}$ as a minor. Branching on every possible subset of $Y_0 \setminus X$, one can guess the intersection I of $\tilde{X}$ with $Y_0 \setminus X$. By Lemma 16, the branching degree is $2^{O(k)}$. As (G, X, k) is a YES-instance, for at least one of the guessed subsets I, the instance $(G_I, Y_0 \setminus I, k - |I|)$ is a YES-instance of the DISJOINT PLANAR-$\mathcal{F}$-DELETION problem. By Lemma 17, the partition $\mathcal{P} = (Y_0 \setminus I) \uplus Y_1 \uplus \cdots \uplus Y_\ell$, where $\{Y_1, \ldots, Y_\ell\}$ is the set of clusters of $G_I - Y_0$, is an $(O(k), r + 2t_{\mathcal{F}})$-protrusion decomposition of G_I. ☐

5.3 Solving Planar-F-Deletion with a linear protrusion decomposition

After having proved Proposition 5, we can now focus in this subsection on solving DISJOINT PLANAR-$\mathcal{F}$-DELETION in single-exponential time when a linear protrusion decomposition is given. Let $P_\Pi(G, S)$ denote an MSO formula (of bounded size) which holds if and only if $G - S$ is $\mathcal{F}$-minor-free.

Consider an instance (G, Y_0, k) of DISJOINT PLANAR-$\mathcal{F}$-DELETION equipped with a linear protrusion decomposition $\mathcal{P}$ of G. Let $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ be an (α, β)-protrusion decomposition of G for some constant β. The key observation is that for every restricted protrusion Y_i, there is a finite number of representatives such that any partial solution lying on Y_i can be replaced with one of these while preserving the feasibility of the solution.

We fix a constant t. Let $\mathcal{U}_t$ be the universe of t-boundaried graphs, and let $\mathcal{U}_t^{\text{small}}$ denote the universe of t-boundaried graphs having a tree-decomposition of width $t - 1$ with all boundary vertices contained in one bag. Throughout this subsection we will assume that all the restricted protrusions belonging to a given protrusion decomposition have the same boundary size, equal to the maximum boundary size over all protrusions. This assumption is licit as if some protrusion has smaller boundary size, we can add dummy independent vertices to it without interfering with the structure of the solutions and without increasing the treewidth.

Definition 14. Let $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ be an (α, t)-protrusion decomposition of G. For each $1 \leqslant i \leqslant \ell$, we define the following equivalence relation $\sim_{\mathcal{F},i}$ on subsets of Y_i: for $Q_1, Q_2 \subseteq Y_i$, we define $Q_1 \sim_{\mathcal{F},i} Q_2$ if for every $H \in \mathcal{U}_t$, $G[Y_i^+ \setminus Q_1] \oplus H$ is $\mathcal{F}$-minor-free if and only if $G[Y_i^+ \setminus Q_2] \oplus H$ is $\mathcal{F}$-minor-free.

Note that G equipped with $\mathcal{P}$ can be viewed as a gluing of two β-boundaried graphs $G[Y_i^+]$ and $G \ominus G[Y_i^+]$, for any $1 \leqslant i \leqslant \ell$, where $Y_i^+ = N_{G_I}[Y_i]$. Let us consider the equivalence relation $\sim_{\mathcal{F},i}$ applied on Y_i when G is viewed as such gluing. Extending the notation suggested in Section 2, we say that $\mathcal{S}$ is a *set of minimum-sized representatives* of the equivalence relation $\approx$ if $\mathcal{S}$ contains exactly one element of minimum cardinality from every equivalence class under $\approx$. Let $\mathcal{R}(Y_i) := \{Q_1^i, \ldots, Q_{q_i}^i\}$ be a set of minimum-sized representatives of equivalence classes under $\sim_{\mathcal{F},i}$ for every $1 \leqslant i \leqslant \ell$. We say that a set $\tilde{X} \subseteq V(G) \setminus Y_0$ is *decomposable* if $\tilde{X} = Q^1 \cup \cdots \cup Q^\ell$ for some $Q^i \in \mathcal{R}(Y_i)$ for $1 \leqslant i \leqslant \ell$.

Lemma 18 (Solution decomposability)**.** *Let (G, Y_0, k) be an instance of* DISJOINT PLANAR-$\mathcal{F}$-DELETION *and let $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ be an (α, β)-protrusion decomposition of G. Then, there* *exists a solution $\tilde{X} \subseteq V(G) \setminus Y_0$ of size at most k if and only if there exists a decomposable solution $\tilde{X}^* \subseteq V(G) \setminus Y_0$ of size at most k.*

Proof. Let $\tilde{X}$ be a subset of $V(G) \setminus Y_0$. Let $S_i := \tilde{X} \cap Y_i$ for every $1 \leqslant i \leqslant \ell$, $\bar{S}_i := \tilde{X} \cap (V(G) \setminus Y_i)$ and let $H := G \ominus G[Y_i^+] - \bar{S}_i$ be the associated t-boundaried graph with $\mathbf{bd}(H) := \mathbf{bd}(G \ominus G[Y_i^+])$. Fix i and choose the (unique) representative $Q^i \in \mathcal{R}(Y_i)$ such that $Q^i \sim_{\mathcal{F},i} S_i$. Note that $S_i \cap \mathbf{bd}(H) = \bar{S}_i \cap \mathbf{bd}(H) = \emptyset$.

We claim that $G - \tilde{X}$ is $\mathcal{F}$-minor-free if and only if $G - (Q^i \cup \bar{S}_i)$ is $\mathcal{F}$-minor-free. Indeed, $G - \tilde{X} = G - (S_i \cup \bar{S}_i)$, which can be written as $G[Y_i^+ \setminus S_i] \oplus H$. From the choice of $Q^i \in \mathcal{R}(Y_i)$ such that $Q^i \sim_{\mathcal{F},i} S_i$, it follows that $G[Y_i^+ \setminus S_i] \oplus H$ is $\mathcal{F}$-minor-free if and only if $G[Y_i^+ \setminus Q^i] \oplus H$ is so. Noting that $G[Y_i^+ \setminus Q^i] \oplus H = G - (Q^i \cup \bar{S}_i)$ proves our claim.

By replacing each S_i with its representative $Q^i \in \mathcal{R}(Y_i)$, we eventually obtain $\tilde{X}^*$ of the form $\tilde{X}^* = \bigcup_{1 \leqslant i \leqslant \ell} Q^i$, where $Q^i \in \mathcal{R}(Y_i)$ is the representative of S_i for every $1 \leqslant i \leqslant \ell$. Finally, it holds that $P_\Pi(G, \tilde{X}^*)$ if and only if $P_\Pi(G, \tilde{X})$. It remains to observe that $|Q^i| \leqslant |S_i|$, as we selected a minimum-sized set of an equivalence class of $\sim_{\mathcal{F},i}$ as its representative. $\square$

We are now ready to prove Proposition 6.

Reminder of Proposition 6. *Let (G, Y_0, k) be an instance of* Disjoint Planar-$\mathcal{F}$-Deletion *and let $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ be an (α, β)-protrusion decomposition of G, for some constant β. There exists an $2^{O(\ell)} \cdot n$-time algorithm that computes a solution $\tilde{X} \subseteq V(G) \setminus Y_0$ of size at most k if it exists, or correctly decides that there is no such solution.*

Proof. By Lemma 18, either there exists a solution of size at most k which is decomposable or (G, Y_0, k) is a No-instance. Henceforth we will just look for decomposable solutions.

We first need to compute, for all $1 \leqslant i \leqslant \ell$, a set of representatives $\mathcal{R}(Y_i)$ of the equivalence relations $\sim_{\mathcal{F},i}$. To that aim, we define an equivalence relation $\equiv_{\mathcal{F},t}$ on $\mathcal{U}_t^{\text{small}}$, with the objective of capturing all possible behaviors of the graphs $G \ominus G[Y_i^+]$ (we call such graphs the *context* of the restricted protrusion Y_i). For two t-boundaried graphs K_1 and K_2 from $\mathcal{U}_t^{\text{small}}$, we say that:

$$K_1 \equiv_{\mathcal{F},t} K_2 \text{ if for every } Y \in \mathcal{U}_t^{small},\ K_1 \oplus Y \text{ is } \mathcal{F}\text{-minor-free iff } K_2 \oplus Y \text{ is } \mathcal{F}\text{-minor-free.}$$

As the property of being $\mathcal{F}$-minor-free is MSO-definable, $\equiv_{\mathcal{F},t}$ has finitely many equivalence classes for each fixed t and there is a finite set $\mathcal{K} = \{K_1, \ldots, K_M\}$ of representatives of $\equiv_{\mathcal{F},t}$ (cf. for instance [22, 32]). Observe that the set $\mathcal{K}$ is independent from the instance (it depends only on the problem). For the sake of readability we now assume that the set $\mathcal{K}$ is given. Note that this assumption would make the proof *non-constructive* and therefore the algorithm *non-uniform* on the family $\mathcal{F}$ (that is, for each family $\mathcal{F}$ we would deduce the existence of a different algorithm). In the paragraph below the end of the proof we briefly explain how this set $\mathcal{K}$ can be efficiently constructed in linear time, yielding a constructive and uniform algorithm.

So given the set $\mathcal{K} = \{K_1, \ldots, K_M\}$, we now proceed to find the set of representatives $\mathcal{R}(Y_i)$ in time $O(|Y_i|)$ for every $1 \leqslant i \leqslant \ell$. Our strategy is inspired by the method of test sets [1]. We consider the set of all binary vectors with M coordinates, to which we give the following interpretation. Each fixed such vector $\mathbf{v} = (b_1, \ldots, b_M)$ will correspond to a minimum-sized subset $Q_{\mathbf{v}} \subseteq Y_i$ such that, for $1 \leqslant j \leqslant M$, the graph $G[(Y_i^+ \setminus Q_{\mathbf{v}})] \oplus K_j$ is $\mathcal{F}$-minor-free iff $b_j = 1$. Formally, for each binary vector $(b_1, \ldots, b_M)$ of length M we consider the following optimization problem: find a set $Q \subseteq Y_i$ of minimum size such that $\varphi(G[Y_i^+], Q)$ holds. Here $\varphi(G[Y_i^+], Q) := (Q \subseteq Y_i) \wedge \left(\bigwedge_{j=1}^{M} \tilde{b}_j\right)$, where $\tilde{b}_j := \varphi_{K_j}(G[Y_i^+], Q)$ if $b_j = 1$ and $\tilde{b}_j := \neg\varphi_{K_j}(G[Y_i^+], Q)$ if $b_j = 0$, each $\varphi_{K_j}(G[Y_i^+], Q)$ stating that $G[(Y_i^+ \setminus Q)] \oplus K_j$ is $\mathcal{F}$-minor-free. For each fixed $K_j \in \mathcal{U}_t^{\text{small}}$, whether $G[(Y_i^+ \setminus Q)] \oplus K_j$ is

$\mathcal{F}$-minor-free or not depends only on $G[Y_i^+]$ and Q, and moreover this property can be expressed as an MSO formula. As φ is an MSO formula, we can apply the linear-time dynamic programming algorithm of Borie *et al.* [14] on graphs of bounded treewidth to solve the associated optimization problem. Note that the running time is $O(|Y_i|)$, whose hidden constant depends solely on $|P_\Pi|$ and the treewidth t.

Claim. *Let $\mathcal{R}_i$ be the set of the optimal solutions over all 2^M binary vectors of length M, obtained as explained above. Then $\mathcal{R}_i$ is a set of minimum-sized representatives of $\sim_{\mathcal{F},i}$.*

Proof. We fix $t := 2t_{\mathcal{F}} + r$, so we can assume that all protrusions Y_i^+ belong to $\mathcal{U}_t^{\text{small}}$. First note that in the definition the equivalence relation $\sim_{\mathcal{F},i}$ (cf. Definition 14), one only needs to consider graphs $H \in \mathcal{U}_t^{\text{small}}$. Indeed, if $H \in \mathcal{U}_t \setminus \mathcal{U}_t^{\text{small}}$, then for any $Q \subseteq Y_i$ it follows that $G[Y_i^+ \setminus Q] \oplus H$ is *not* $\mathcal{F}$-minor-free (as $\mathbf{tw}(G[Y_i^+ \setminus Q] \oplus H) \geqslant 2t_{\mathcal{F}} + r > t_{\mathcal{F}}$), so in order to define the equivalence classes of subsets of Y_i it is enough to consider $H \in \mathcal{U}_t^{\text{small}}$. In other words, only the elements of $\mathcal{U}_t^{\text{small}}$ can *distinguish* the subsets of Y_i with respect to $\sim_{\mathcal{F},i}$.

Let $Q \subseteq Y_i$, and we want to prove that there exists $R_Q \in \mathcal{R}_i$ such that $Q \sim_{\mathcal{F},i} R_Q$, that is, such that for any $H \in \mathcal{U}_t$, $G[Y_i^+ \setminus Q] \oplus H$ is $\mathcal{F}$-minor-free iff $G[Y_i^+ \setminus R_Q] \oplus H$ is $\mathcal{F}$-minor-free. By the remark in the above paragraph, we can assume that $H \in \mathcal{U}_t^{\text{small}}$, as otherwise the statement is trivially true. Let $\mathbf{v}_Q = (b_1, \ldots, b_M)$ be the binary vector on M coordinates such that, for $1 \leqslant j \leqslant M$, $b_j = 1$ iff $G[(Y_i^+ \setminus Q) \oplus K_j]$ is $\mathcal{F}$-minor-free. We define R_Q to be the graph in $\mathcal{R}_i$ corresponding to the vector $\mathbf{v}_Q$. As $H \in \mathcal{U}_t^{\text{small}}$, there exists $K_H \in \mathcal{K}$ such that $H \equiv_{\mathcal{F},t} K_H$. Then, $G[Y_i^+ \setminus Q] \oplus H$ is $\mathcal{F}$-minor-free iff $G[Y_i^+ \setminus Q] \oplus K_H$ is $\mathcal{F}$-minor-free, which by construction is $\mathcal{F}$-minor-free iff $G[Y_i^+ \setminus R_Q] \oplus K_H$ is $\mathcal{F}$-minor-free, which is in turn $\mathcal{F}$-minor-free iff $G[Y_i^+ \setminus R_Q] \oplus H$ is $\mathcal{F}$-minor-free, as we wanted to prove. □

To summarize the above discussion, a set of minimum-sized representatives $\mathcal{R}(Y_i)$ can be constructed in time $O(|Y_i|)$, and therefore all the sets of such representatives can be constructed in time $O(n)$.

Once the sets $\mathcal{R}(Y_i)$ of representatives for the equivalence relations $\sim_{\mathcal{F},i}$ have been computed for all $1 \leqslant i \leqslant \ell$, it remains to test every possible decomposable set $\tilde{X}$ (see Lemma 18). Since we have computed a *minimum-sized* set of representatives $\mathcal{R}(Y_i)$ for each $\sim_{\mathcal{F},i}$, it follows that there exists a solution $\tilde{X} \subseteq V(G) \setminus Y_0$ of size at most k if and only if there exists a decomposable solution $\tilde{X}^* \subseteq V(G) \setminus Y_0$ of size at most k which is made of representatives from the sets $\mathcal{R}(Y_i)$. Observe that for a given decomposable set $\tilde{X}$, one can decide if $\tilde{X}$ is a solution or not in time $O(h(t_{\mathcal{F}}) \cdot n)$. Indeed, for $\tilde{X}$ to be a solution, the treewidth of $G - \tilde{X}$ is at most $t_{\mathcal{F}} - 1$. Using the algorithm of Bodlaender [7], one can decide in time $2^{O(t_{\mathcal{F}}^3)} \cdot n$ whether a graph is of treewidth at most $t_{\mathcal{F}}$ and if so, build a tree-decomposition of width at most $t_{\mathcal{F}}$. Courcelle's theorem [21] says that testing an MSO-definable property on treewidth-$t_{\mathcal{F}}$ graphs can be done in linear time, where the hidden constant depends solely on the treewidth $t_{\mathcal{F}}$ and the length of the MSO-sentence. It follows that one can decide whether $G - \tilde{X}$ is $\mathcal{F}$-minor-free or not in time $O(h(t_{\mathcal{F}}) \cdot n)$. Here $h(t_{\mathcal{F}})$ is an additive function resulting from Bodlaender's treewidth testing algorithm and Courcelle's MSO-model checking algorithm, which depends solely on the treewidth $t_{\mathcal{F}}$ and the MSO formula $|P_\Pi(G, \tilde{X})|$. It remains to observe that there are at most $2^{O(\ell)}$ decomposable sets to consider. This is because an MSO-definable graph property has finitely many equivalence classes on $\mathcal{U}_t$ for every fixed t [12, 21] (hence, also on $\mathcal{U}_t^{\text{small}}$), and being $\mathcal{F}$-minor-free is an MSO-definable property. □

Constructing the sets of representatives for $\equiv_{\mathcal{F},t}$. Let $\Phi_{\mathcal{F}}$ be the MSO-formula expressing that a graph is $\mathcal{F}$-minor free. The set $\mathcal{K}$ of representatives can be efficiently constructed on the universe $\mathcal{U}_t^{\text{small}}$ from $\Phi_{\mathcal{F}}$ and the boundary size t. Let us discuss the main line of the proof of this fact. Courcelle's theorem is proved[10] by converting an MSO formula φ on tree-decompositions of width t into another MSO formula φ' on labeled trees. Trees, in which every internal node has bounded *fan-in* and every node is labeled with an alphabet chosen from a fixed set, are considered as a *tree language*, which is a natural generalization of the usual string language. It is well-known (as the analogue of Büchi's theorem [15] on tree languages) that the set of labeled trees for which an MSO formula holds form a regular (tree) language[11]. Moreover, based on its proof it is not difficult to construct a finite tree automaton (cf. for instance [35]). In particular, the number of states in the corresponding tree automaton is bounded by a constant depending only on φ and t. From this, it is possible to prove (using a "tree" pumping lemma) that one can assume that the height of a *distinguishing extension* of two labeled trees is bounded by a constant as well (in fact, the size of the tree automaton). Hence we can enumerate all possible labeled trees of bounded height, which will be a test set to construct the set of representatives $\mathcal{K}$. Now one can apply the so-called *method of test sets* (basically implicit in the proof of the Myhill-Nerode theorem [70], see [1, 32] for more details) and retrieve the set of representatives $\mathcal{K}$. Finally, the reader can check that each of these standard procedures can be implemented in time $O(n)$, which implies that the algorithm of Proposition 6 has overall running time $2^{O(\ell)} \cdot n$. (We note that an approach similar to the one described here can be found in [84, Corollary 3.13].)

5.4 Proof of Theorem II

We finally have all the ingredients to prove Theorem II.

Reminder of Theorem II. *The parameterized* Planar-$\mathcal{F}$-Deletion *problem can be solved in time* $2^{O(k)} \cdot n^2$.

Proof. Lemma 14 states that Planar-$\mathcal{F}$-Deletion can be reduced to Disjoint Planar-$\mathcal{F}$-Deletion so that the former can be solved in single-exponential time solvable provided that the latter is so, and the degree of the polynomial function just increases by one. We now proceed to solve Disjoint Planar-$\mathcal{F}$-Deletion in time $2^{O(k)} \cdot n$. Given an instance (G, X, k) of Disjoint Planar-$\mathcal{F}$-Deletion, we apply Proposition 5 to either correctly decide that (G, X, k) is a No-instance, or identify in time $2^{O(k)} \cdot n$ a set $I \subseteq V(G)$ of size at most k and a $(O(k), 2t_{\mathcal{F}}+r)$-protrusion decomposition $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ of $G - I$, with $X \subseteq Y_0$, such that there exists a set $X' \subseteq V(G) \setminus Y_0$ of size at most $k - |I|$ such that $G - \tilde{X}$, with $\tilde{X} = X' \cup I$, is H-minor-free for every graph $H \in \mathcal{F}$. Finally, using Proposition 6 we can solve the instance $(G_I, Y_0 \setminus I, k - |I|)$ in time $2^{O(k)} \cdot n$. ☐

6 Conclusions and further research

We presented a simple algorithm to compute protrusion decompositions for graphs G that come equipped with a set $X \subseteq V(G)$ such that the treewidth of $G - X$ is at most some fixed constant t.

[10] In fact, there is more than one proof of Courcelle's theorem. The one we depict in this article is as presented in [35], which differs from the original proof of Courcelle [21].

[11] A regular tree language is an analogue of a regular language on labeled trees. Appropriately defined, most of the nice properties on string regular languages transfer immediately to tree regular languages. It is beyond the scope of this article to give details about tree languages and tree automatons. We invite the interested readers to [22, 32].

Then we showed that this algorithm can be used in order to achieve two different sets of results: linear kernels on graphs excluding a fixed topological minor, and a single-exponential parameterized algorithm for the Planar-$\mathcal{F}$-Deletion problem.

Concerning our kernelization algorithm, the first main question is whether similar results can be obtained for an even larger class of (sparse) graphs. A natural candidate is the class of graphs of bounded expansion (see [73] for the definition), which strictly contains H-topological-minor-free graphs. Let us now argue that the existence of linear kernels for some of the considered problems on graphs of bounded expansion seems to be as plausible as on general graphs. Indeed, consider for instance the Treewidth-t Vertex Deletion problem, which is clearly treewidth-bounding. Take a general graph G on n vertices as input of Treewidth-t Vertex Deletion, and let G' be the graph obtained from G by subdividing each edge n times. Note that G' has bounded expansion, and that this operation does not increase the treewidth. As far as $t \geqslant 1$, it is easy to see that one can assume that none of the newly added vertices belong to a solution, and thus the size of an optimal solution is the same in G and G'. Therefore, obtaining a kernel for Treewidth-t Vertex Deletion on graphs of bounded expansion is as hard as on general graphs. According to Fomin *et al.* [41], this problem has a kernel of size $k^{O(t)}$ on general graphs, and no uniform polynomial kernel (that is, a polynomial kernel whose degree does not depend on t) is known. Since graphs of bounded expansion strictly contain H-topological-minor-free graphs, and there are not well-known graph classes in between, our kernelization result may settle the limit of meta-theorems about the existence of linear, or even uniform polynomial, kernels on sparse graph classes.

The second main question is which other problems have linear kernels on H-topological-minor-free graphs. In particular, it has been recently proved by Fomin *et al.* that (Connected) Dominating Set has a linear kernel on H-minor-free graphs [43] and on H-topological-minor-free graphs [36]. It would be interesting to investigate how the structure theorem by Grohe and Marx [48] can be used in this context.

We would like to note that the degree of the polynomial of the running time of our kernelization algorithm depends linearly on the size of the excluded topological minor H. It seems that the recent *fast protrusion replacer* of Fomin *et al.* [40] could be applied to get rid of the dependency on H of the running time.

Let us now discuss some further research related to our single-exponential algorithm for Planar-$\mathcal{F}$-Deletion. As mentioned in the introduction, no single-exponential algorithm is known when the family $\mathcal{F}$ does not contain any planar graph. Is it possible to find such a family, or can it be proved that, under some complexity assumption, a single-exponential algorithm is not possible? See [52] for recent advances in this direction. An ambitious goal would be to optimize the constants involved in the function $2^{O(k)}$, possibly depending on the family $\mathcal{F}$, and maybe even proving lower bounds for such constants, in the spirit of Lokshtanov *et al.* [63] for problems parameterized by treewidth.

It also makes sense to forbid the family of graphs $\mathcal{F}$ according to another containment relation, like topological minor. Using the fact that if H is a graph with maximum degree at most 3, a graph G contains H as a minor if and only if G contains H as a topological minor (hence, graphs that exclude a planar graph H with maximum degree at most 3 as a topological minor have bounded treewidth), it can be proved that our approach also yields a single-exponential algorithm for Planar-Topological-$\mathcal{F}$-Deletion as far as $\mathcal{F}$ contains some planar (connected or disconnected) graph with maximum degree at most 3.

We showed (in Section 5.3) how to obtain single-exponential algorithms for Disjoint Planar-$\mathcal{F}$-Deletion with a given linear protrusion decomposition. This approach seems to be applicable to general vertex deletion problems to attain a property expressible in CMSO (but probably, the fact of having bounded treewidth is needed in order to make the algorithm constructive). It would be interesting to generalize this technique to p-MAX-CMSO or p-EQ-CMSO problems, as well as to edge subset problems.

Recently, a randomized (Monte Carlo) constant-factor approximation algorithm for PLANAR-$\mathcal{F}$-DELETION has been given by Fomin *et al.* [40]. Finding a deterministic constant-factor approximation remains open. Also, the existence of linear or polynomial kernels for PLANAR-$\mathcal{F}$-DELETION, or even for the general $\mathcal{F}$-DELETION problem, is an exciting avenue for further research. It seems that significant advances in this direction, namely for PLANAR-$\mathcal{F}$-DELETION, have been done by Fomin *et al.* [41]. It is worth mentioning that recently Fomin *et al.* [37] have proved that $\mathcal{F}$-DELETION admits a polynomial kernel when parameterized by the size of a vertex cover.

The running time of the algorithm given in Theorem II is $2^{O(k)} \cdot n^2$, which can be improved to $2^{O(k)} \cdot n \log^2 n$ by using the following trick based on [40]. Let t be a bound on the treewidth of any graph excluding a planar graph in $\mathcal{F}$ as a minor, and let G the the input graph to our problem. Instead of doing iterative compression, we first solve on G in time $2^{O(k)} \cdot n \log^2 n$ the problem consisting on deleting at most k vertices in order to obtain a graph of treewidth at most t, using the algorithm of [40] (note that we can do so, as all obstructions for treewidth are clearly connected). If we fail, we know that G is a negative instance of PLANAR-$\mathcal{F}$-DELETION, and we are done. Otherwise, let X be such a set of at most k vertices. Now we can solve PLANAR-$\mathcal{F}$-DELETION on the graph $G - D$ in time $O(n)$, as it has bounded treewidth, and obtain a set X' of size at most k such that $G - (X \cup X')$ is $\mathcal{F}$-minor-free. Finally, we can guess the intersection of the set $X \cup X'$ with the solution of PLANAR-$\mathcal{F}$-DELETION, and then solve the DISJOINT PLANAR-$\mathcal{F}$-DELETION problem in time $2^{O(k)} \cdot n$, as it is done in Section 5.

In the parameterized dual version of the $\mathcal{F}$-DELETION problem, the objective is to find at least k vertex-disjoint subgraphs of an input graph, each of them containing some graph in $\mathcal{F}$ as a minor. For $\mathcal{F} = \{K_3\}$, the problem corresponds to k-DISJOINT CYCLE PACKING, which does not admit a polynomial kernel on general graphs [11] unless co-NP $\subseteq$ NP/*poly*. Does this problem, for some non-trivial choice of $\mathcal{F}$, admit a single-exponential parameterized algorithm?

Acknowledgements. We thank Dimitrios M. Thilikos, Bruno Courcelle, Daniel Lokshtanov, and Saket Saurabh for interesting discussions and helpful remarks on the manuscript. Our gratitude also goes to the anonymous reviewers for thorough comments that improved the presentation of the article.

References

[1] K. R. Abrahamson and M. R. Fellows. Finite automata, bounded treewidth, and well-quasiordering. In *Proc. of Graph Structure Theory, Contemporary Mathematics 147*, pages 539–564. American Mathematical Society, 1991.

[2] I. Adler, F. Dorn, F. V. Fomin, I. Sau, and D. M. Thilikos. Faster parameterized algorithms for minor containment. *Theoretical Computer Science*, 412(50):7018–7028, 2011.

[3] J. Alber, M. R. Fellows, and R. Niedermeier. Polynomial-time data reduction for Dominating Set. *Journal of the ACM*, 51:363–384, 2004.

[4] S. Arnborg, B. Courcelle, A. Proskurowski, and D. Seese. An algebraic theory of graph reduction. *Journal of the ACM*, 40(5):1134–1164, 1993.

[5] J. Battle, F. Harary, Y. Kadama, and J. Youngs. Additivity of the genus of a graph. *Bulletin of the American Mathematical Society*, 68:565–568, 1962.

[6] H. L. Bodlaender. Dynamic programming on graphs with bounded treewidth. In *Proc. of 15th ICALP*, number 317 in LNCS, pages 105–118. Springer, 1988.

[7] H. L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25:1305–1317, 1996.

[8] H. L. Bodlaender, F. V. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh, and D. M. Thilikos. (Meta) Kernelization. In *Proc. of 50th FOCS*, pages 629–638. IEEE Computer Society, 2009.

[9] H. L. Bodlaender and E. Penninkx. A linear kernel for Planar Feedback Vertex Set. In *Proc. of 3rd IWPEC*, number 5018 in LNCS, pages 160–171. Springer, 2008.

[10] H. L. Bodlaender, E. Penninkx, and R. B. Tan. A linear kernel for the k-Disjoint Cycle problem on planar graphs. In *Proc. of 19th ISAAC*, number 5369 in LNCS, pages 306–317. Springer, 2008.

[11] H. L. Bodlaender, S. Thomassé, and A. Yeo. Kernel bounds for disjoint cycles and disjoint paths. In *Proc. of 17th ESA*, number 5757 in LNCS, pages 635–646. Springer, 2009.

[12] H. L. Bodlaender and B. van Antwerpen-de Fluiter. Reduction algorithms for graphs of small treewidth. *Information and Computation*, 167(2):86–119, 2001.

[13] B. Bollobás and A. Thomason. Proof of a conjecture of Mader, Erdős and Hajnal on topological complete subgraphs. *European Journal of Combinatorics*, 19(8):883–887, 1998.

[14] R. B. Borie, R. G. Parker, and C. A. Tovey. Automatic Generation of Linear-Time Algorithms from Predicate Calculus Descriptions of Problems on Recursively Constructed Graph Families. *Algorithmica*, 7(1):555–581, 1992.

[15] J. R. Büchi. On a Decision Method in Restricted Second-Order Arithmetic. In *Proc. of International Congress on Logic, Methodology, and Philosophy of Science*, pages 1–11, 1962.

[16] L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58:171–176, 1996.

[17] Y. Cao. Linear Recognition of Almost Interval Graphs. *CoRR*, abs/1403.1515, 2014.

[18] Y. Cao and D. Marx. Interval Deletion Is Fixed-Parameter Tractable. *ACM Transactions on Algorithms*, 11(3):1–35, 2015.

[19] J. Chen, H. Fernau, I. Kanj, and G. Xia. Parametric duality and kernelization: lower bounds and upper bounds on kernel size. *SIAM Journal on Computing*, 37:1077–1106, 2007.

[20] J. Chen, F. V. Fomin, Y. Liu, S. Lu, and Y. Villanger. Improved algorithms for feedback vertex set problems. *Journal of Computer and System Sciences*, 74(7):1188–1198, 2008.

[21] B. Courcelle. The Monadic Second-Order Theory of Graphs. I. Recognizable Sets of Finite graphs. *Information and Computation*, 85:12–75, 1990.

[22] B. Courcelle and J. Engelfriet. *Graph Structure and Monadic Second-Order Logic: A Language Theoretic Approach.* Number 138 in Encyclopedia of Mathematics and its Applications. Cambridge University Press, June 2012.

[23] M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. M. M. van Rooij, and J. O. Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *Proc. of 52nd FOCS*, pages 150–159. IEEE Computer Society, 2011.

[24] M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. Wojtaszczyk. An improved FPT algorithm and quadratic kernel for pathwidth one vertex deletion. In *Proc. of 5th IWPEC*, number 6478 in LNCS, pages 95–106. Springer, 2010.

[25] A. Dawar, M. Grohe, and S. Kreutzer. Locally excluding a minor. In *Proc. of 22 LICS*, pages 270–279. IEEE Computer Society, 2007.

[26] B. de Fluiter. *Algorithms for Graphs of Small Treewidth.* PhD thesis, Utrecht University, 1997.

[27] F. K. H. A. Dehne, M. R. Fellows, M. A. Langston, F. A. Rosamond, and K. Stevens. An $O(2^{O(k)}n^3)$ FPT algorithm for the undirected feedback vertex set problem. In *Proc. of 11th COCOON*, number 3595 in LNCS, pages 859–869. Springer, 2005.

[28] R. Diestel. *Graph Theory.* Springer-Verlag, Heidelberg, 4th edition, 2010.

[29] M. J. Dinneen. Too many minor order obstructions. *Journal of Universal Computer Science*, 3(11):1199–1206, 1997.

[30] M. Dom, D. Lokshtanov, and S. Saurabh. Incompressibility through colors and IDs. In *Proc. of 36th ICALP*, number 5555 in LNCS, pages 378–389. Springer, 2009.

[31] F. Dorn, F. V. Fomin, and D. M. Thilikos. Catalan structures and dynamic programming in H-minor-free graphs. In *Proc. of 19th SODA*, pages 631–640. SIAM, 2008.

[32] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer-Verlag, 1999.

[33] M. R. Fellows and M. A. Langston. Nonconstructive tools for proving polynomial-time decidability. *Journal of the ACM*, 35:727–739, 1988.

[34] M. R. Fellows and M. A. Langston. An Analogue of the Myhill-Nerode Theorem and Its Use in Computing Finite-Basis Characterizations (Extended Abstract). In *Proc. of 30th FOCS*, pages 520–525, 1989.

[35] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer-Verlag, 2006.

[36] F. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos. Linear kernels for (connected) dominating set on graphs with excluded topological subgraphs. In *Proc. of 30th STACS*, volume 20 of *LIPIcs*, pages 92–103. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2013.

[37] F. V. Fomin, B. M. P. Jansen, and M. Pilipczuk. Preprocessing Subgraph and Minor Problems: When Does a Small Vertex Cover Help? In *Proc. of 7th IPEC*, number 7535 in LNCS, pages 97–108. Springer, 2012.

[38] F. V. Fomin, D. Lokshtanov, N. Misra, G. Philip, and S. Saurabh. Hitting forbidden minors: Approximation and kernelization. In *Proc. of 28th STACS*, volume 9 of *LIPIcs*, pages 189–200. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2011.

[39] F. V. Fomin, D. Lokshtanov, N. Misra, and S. Saurabh. Nearly optimal FPT algorithms for Planar-$\mathcal{F}$-Deletion. Unpublished manuscript, 2011.

[40] F. V. Fomin, D. Lokshtanov, N. Misra, and S. Saurabh. Planar $\mathcal{F}$-Deletion: Approximation and Optimal FPT Algorithms. In *Proc. of 53rd FOCS*, pages 470-479, IEEE Computer Society, 2012.

[41] F. V. Fomin, D. Lokshtanov, N. Misra, and S. Saurabh. Planar-$\mathcal{F}$ Deletion: Kernelization and obstruction bounds. Unpublished manuscript, 2012.

[42] F. V. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos. Bidimensionality and kernels. In *Proc. of 21st SODA*, pages 503–510. SIAM, 2010.

[43] F. V. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos. Linear kernels for (Connected) Dominating Set on H-minor-free graphs. In *Proc. of 23rd SODA*, pages 82–93. SIAM, 2012.

[44] F. V. Fomin, S. Oum, and D. M. Thilikos. Rank-width and tree-width of H-minor-free graphs. *European Journal of Combinatorics*, 31(7):1617–1628, 2010.

[45] M. Frick and M. Grohe. The complexity of first-order and monadic second-order logic revisited. *Annals of Pure and Applied Logic*, 130(1–3):3–31, 2004.

[46] J. Gajarský, P. Hlinený, J. Obdrzálek, S. Ordyniak, F. Reidl, P. Rossmanith, F. Sánchez Villaamil, and S. Sikdar. Kernelization using structural parameters on sparse graph classes. In *Proc. of 21st ESA*, number 8125 in LNCS, pages 529–540. Springer, 2013.

[47] R. Ganian, F. Slivovsky, and S. Szeider. Meta-kernelization with structural parameters. In *Proc. of 38th MFCS*, number 8087 in LNCS, pages 457–468. Springer, 2013.

[48] M. Grohe and D. Marx. Structure theorem and isomorphism test for graphs with excluded topological subgraphs. *CoRR*, abs/1111.1109, 2011.

[49] J. Guo, J. Gramm, F. Hüffner, R. Niedermeier, and S. Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. *Journal of Computing and System Sciences*, 72(8):1386–1396, 2006.

[50] J. Guo and R. Niedermeier. Linear problem kernels for NP-hard problems on planar graphs. In *Proc. of 34th ICALP*, number 4596 in LNCS, pages 375–386. Springer, 2007.

[51] J. Guo, R. Niedermeier, and S. Wernicke. Fixed-parameter tractability results for full-degree spanning tree and its dual. *Networks*, 56(2):116–130, 2010.

[52] B. M. P. Jansen, D. Lokshtanov, and S. Saurabh. A near-optimal planarization algorithm. In *Proc. of 25th SODA*, pages 1802–1811. SIAM, 2014.

[53] G. Joret, C. Paul, I. Sau, S. Saurabh, and S. Thomassé. Hitting and harvesting pumpkins. In *Proc. of 19th ESA*, number 6942 in LNCS, pages 394–407. Springer, 2011.

[54] I. A. Kanj, M. J. Pelsmajer, M. Schaefer, and G. Xia. On the Induced Matching problem. *Journal of Computer and System Sciences*, 77(6):1058–1070, 2011.

[55] K. Kawarabayashi. Planarity allowing few error vertices in linear time. In *Proc. of 50th FOCS*, pages 639–648. IEEE Computer Society, 2009.

[56] E. J. Kim, C. Paul, and G. Philip. A single-exponential FPT-algorithm for K_4-minor cover problem. In *Proc. of 13th SWAT*, number 7357 in LNCS. Springer, 2012. To appear.

[57] T. Kloks. *Treewidth, Computations and Approximations*. Number 842 in LNCS. Springer, 1994.

[58] J. Komlós and E. Szemerédi. Topological cliques in graphs II. *Combinatorics, Probability and Computing*, 5(01):79–90, 1996.

[59] S. Kreutzer. Algorithmic meta-theorems. *Electronic Colloquium on Computational Complexity*, 16:147, 2009.

[60] S. Kreutzer and S. Tazari. On brambles, grid-like minors, and parameterized intractability of monadic second-order logic. In *Proc. of 21st SODA*, pages 354–364. SIAM, 2010.

[61] J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.

[62] D. Lokshtanov. Wheel-free deletion is $W[2]$-hard. In *Proc. of 3rd IWPEC*, number 5018 in LNCS, pages 141–147. Springer, 2008.

[63] D. Lokshtanov, D. Marx, and S. Saurabh. Slightly superexponential parameterized problems. In *Proc. of 22nd SODA*, pages 760–776. SIAM, 2011.

[64] D. Lokshtanov, M. Mnich, and S. Saurabh. A linear kernel for Planar Connected Dominating Set. *Theoretical Computer Science*, 412(23):2536–2543, 2011.

[65] D. Lokshtanov, S. Saurabh, and S. Sikdar. Simpler parameterized algorithm for OCT. In *Proc. of 20th IWOCA*, number 5874 in LNCS, pages 380–384. Springer, 2009.

[66] D. Marx. Chordal Deletion is fixed-parameter tractable. *Algorithmica*, 57(4):747–768, 2010.

[67] D. Marx and I. Schlotter. Obtaining a planar graph by vertex deletion. *Algorithmica*, 62(3–4):807–822, 2012.

[68] B. Mohar. An obstruction to embedding graphs on surfaces. *Discrete Mathematics*, 78(1–2):135–142, 1989.

[69] H. Moser and S. Sikdar. The parameterized complexity of the Induced Matching problem. *Discrete Applied Mathematics*, 157(4):715–727, 2009.

[70] A Nerode. Linear Automaton Transformations. *Proceedings of the American Mathematical Society*, 9:541–544, 1958.

[71] J. Nešetřil and P. Ossona de Mendez. On nowhere dense graphs. *European Journal of Combinatorics*, 32(4):600–617, 2011.

[72] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[73] J. Něšětril and P. Ossona de Mendez. Grad and classes with bounded expansion I. Decompositions. *European Journal of Combinatorics*, 29(3), 2008.

[74] G. Philip, V. Raman, and Y. Villanger. A quartic kernel for Pathwidth-One Vertex Deletion. In *Proc. of 36th WG*, number 6410 in LNCS, pages 196–207. Springer, 2010.

[75] Arash Rafiey. Single Exponential FPT Algorithm for Interval Vertex Deletion and Interval Completion Problem. *CoRR*, abs/1211.4629, 2012.

[76] B. A. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004.

[77] N. Robertson and P. D. Seymour. Graph Minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7:309–322, 1986.

[78] N. Robertson and P. D. Seymour. Graph Minors. V. Excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41:92–114, 1986.

[79] N. Robertson and P. D. Seymour. Graph Minors. X. Obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52:153–190, 1991.

[80] N. Robertson and P. D. Seymour. Graph Minors. XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63:65–110, 1995.

[81] N. Robertson and P. D. Seymour. Graph Minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92:325–357, 2004.

[82] J. Rué, I. Sau, and D. M. Thilikos. Dynamic programming for graphs on surfaces. *CoRR*, abs/1104.2486, 2011. To appear in ACM Transactions on Algorithms.

[83] A. Thomason. The extremal function for complete minors. *Journal of Combinatorial Theory, Series B*, 81(2):318–338, 2001.

[84] R. van Bevern, M. R. Fellows, S. Gaspers, and F. A. Rosamond. Myhill-Nerode Methods for Hypergraphs. *Algorithmica*, pages 1–34, 2015. DOI `10.1007/s00453-015-9977-x`.

[85] M. Yannakakis and F. Gavril. Edge dominating sets in graphs. *SIAM Journal of Applied Mathematics*, 38(3):364–372, 1980.

A Edge modification problems are not minor-closed

A graph problem Π is *minor-closed* if whenever G is a YES-instance of Π and G' is a minor of G, then G' is also a YES-instance of Π. It is easy to see that $\mathcal{F}$-(VERTEX-)DELETION is minor-closed, and therefore it is FPT by Robertson and Seymour [80]. Here we show that the edge modification versions, namely, $\mathcal{F}$-EDGE-CONTRACTION and $\mathcal{F}$-EDGE-REMOVAL (defined in the natural way), are not minor-closed.

Edge contraction. In this case, the problem Π is whether one can contract at most k edges from a given graph G so that the resulting graph does not contain any of the graphs in $\mathcal{F}$ as a minor. Let $\mathcal{F} = \{K_5, K_{3,3}\}$, and let G be the graph obtained from K_5 by subdividing every edge k times, and adding an edge e between two arbitrary original vertices of K_5. Then G can be made planar just by contracting edge e, but if G' is the graph obtained from G by deleting e (which is a minor of G), then at least $k+1$ edge contractions are required to make G' planar.

Edge deletion. In this case, the problem Π is whether one can delete at most k edges from a given graph G so that the resulting graph does not contain any of the graphs in $\mathcal{F}$ as a minor. Let G, G', and H be the graphs depicted in Figure 4, and let $k = 1$. Then G can be made H-minor-free by deleting edge e, but G', which is the graph obtained from G by contracting edge e, needs at least two edge deletions to be H-minor-free.

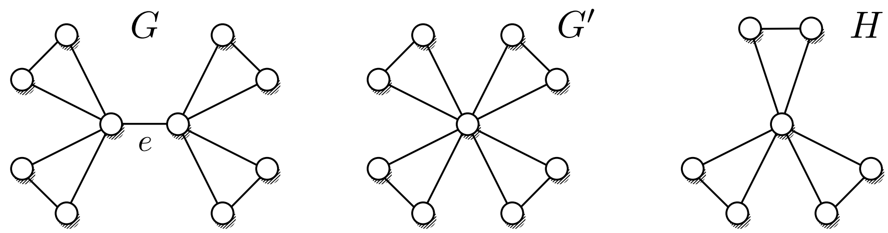

Figure 4: Example to show that $\mathcal{F}$-EDGE-REMOVAL is not minor-closed.

B Disconnected planar obstructions

Let us argue that there exist natural obstruction sets that contain disconnected planar graphs. Following Dinneen [29], given an integer $\ell \geqslant 0$ and a graph invariant function λ that maps graphs to integers such that whenever $H \preceq_m G$ we also have $\lambda(H) \leqslant \lambda(G)$, we say that the graph class $\mathcal{G}_\lambda^\ell := \{G : \lambda(G) \leqslant \ell\}$ is an *ℓ-parameterized lower ideal.* By Robertson and Seymour [80], we know that for each ℓ-parameterized lower ideal $\mathcal{G}_\lambda^\ell$ there exists a finite graph family $\mathcal{F}$ such that $\mathcal{G}_\lambda^\ell$ has precisely $\mathcal{F}$ as (minor) obstruction set. In this setting, the $\mathcal{F}$-DELETION problem (parameterized by k) asks whether k vertices can be removed from a graph G so that the resulting graph belongs to the corresponding ℓ-parameterized lower ideal $\mathcal{G}_\lambda^\ell$. For instance, the parameterized FEEDBACK VERTEX SET problem corresponds to the 0-parameterized lower ideal with graph invariant **fvs**, namely $\mathcal{G}_{\mathbf{fvs}}^0$, which is characterized by $\mathcal{F} = \{K_3\}$ and therefore $\mathcal{G}_{\mathbf{fvs}}^0$ is the set of all forests. Interestingly, it is proved in [29] that for $\ell \geqslant 1$, the obstruction set of many interesting graph invariants (such as ℓ-VERTEX COVER, ℓ-FEEDBACK VERTEX SET, or ℓ-FACE COVER to name just a few) contains the disjoint union of obstructions for $\ell - 1$. As for the above-mentioned problems there is a planar obstruction for $\ell = 0$, we conclude that for $\ell \geqslant 1$ the corresponding family $\mathcal{F}$ contains *disconnected* planar obstructions.

C Disconnected Planar-$\mathcal{F}$-Deletion has not finite integer index

We proceed to prove that if $\mathcal{F}$ is a family of graphs containing some disconnected graph H (planar or non-planar), then the $\mathcal{F}$-Deletion problem has not finite integer index (FII) in general.

We shall use the equivalent definition of FII as suggested for graph optimization problems, see [26]. For a graph problem o-Π, the equivalence relation $\sim_{o\text{-}\Pi,t}$ on t-boundaried graphs is defined as follows. Let G_1 and G_2 be two t-boundaried graphs. We define $G_1 \sim_{\Pi,t} G_2$ if and only if there exists an integer i such that for any t-boundaried graph H, it holds $\pi(G_1 \oplus H) = \pi(G_2 \oplus H) + i$, where $\pi(G)$ denotes the optimal value of problem o-Π on graph G. We claim that $G_1 \sim_{\Pi,t} G_2$ if and only if $G_1 \equiv_{\Pi,t} G_2$ (recall Definition 8 of canonical equivalence), where Π is the parameterized version of o-Π with the solution size as the parameter. Suppose $G_1 \sim_{\Pi,t} G_2$ and let $\pi(G_1 \oplus H) = \pi(G_2 \oplus H) + i$. Then

$$(G_1 \oplus H, k) \in \Pi \;\Leftrightarrow\; \pi(G_1 \oplus H) \leqslant k \;\Leftrightarrow\; \pi(G_2 \oplus H) \leqslant k - i(G_2 \oplus H, k - i) \in \Pi,$$

and thus the forward implication holds. The opposite direction is easy to see.

Let F_1 and F_2 be two incomparable graphs with respect to the minor relation, and let F be the disjoint union of F_1 and F_2. For instance, if we want F to be planar, we can take $F_1 = K_4$ and $F_2 = K_{2,3}$. We set $\mathcal{F} = \{F\}$. Let Π be the non-parameterized version of $\mathcal{F}$-Vertex Deletion.

For $i \geqslant 1$, let G_i be the 1-boundaried graph consisting of the boundary vertex v together with i disjoint copies of F_1, and for each such copy, we add and edge between v and an arbitrary vertex of F_1. Similarly, for $j \geqslant 1$, let H_j be the 1-boundaried graph consisting of the boundary vertex u together with j disjoint copies of F_2, and for each such copy, we add and edge between u and an arbitrary vertex of F_2.

By construction, if $i, j \geqslant 1$, it holds $\pi(G_i \oplus H_j) = \min\{i, j\}$. Then, if we take $1 \leqslant n < m$,

$$\begin{aligned} \pi(G_n \oplus H_{n-1}) - \pi(G_m \oplus H_{n-1}) &= (n-1) - (n-1) \;=\; 0, \\ \pi(G_n \oplus H_m) - \pi(G_m \oplus H_m) &= n - m \;<\; 0. \end{aligned}$$

Therefore, G_n and G_m do not belong to the same equivalence class of $\sim_{\Pi,1}$ whenever $1 \leqslant n < m$, so $\sim_{\Pi,1}$ has infinitely many equivalence classes, and thus Π has not FII.

In particular, the above example shows that if $\mathcal{F}$ may contain some disconnected planar graph H, then Planar-$\mathcal{F}$-Deletion has not FII in general.

D MSO formula for topological minor containment

For a fixed graph H we describe and MSO_1-formula Φ_H over the usual structure consisting of the universe $V(G)$ and a binary symmetric relation $\textsc{adj}$ modeling $E(G)$ such that $G \models \Phi_H$ iff

$H \preceq_{tm} G$.

$$
\begin{aligned}
\Phi_H(G) \quad &:= \quad \exists x_{v_1} \ldots \exists x_{v_r} \exists D_{e_1} \ldots \exists D_{e_\ell} \\
& \quad\; \Big(\bigwedge_{1 \leqslant i < j \leqslant r} x_{v_i} \neq x_{v_j} \wedge \bigwedge_{\substack{1 \leqslant i \leqslant r \\ 1 \leqslant j \leqslant \ell}} x_{v_i} \notin D_{e_j} \wedge \bigwedge_{1 \leqslant i < j \leqslant r} \textsc{dis}(D_{e_i}, D_{e_j}) \wedge \bigwedge_{\substack{1 \leqslant j \leqslant \ell \\ e_j = v_i v_k}} \textsc{conn}(x_{v_i}, D_{e_j}, x_{v_k}) \Big) \\
\text{with } \textsc{dis}(X, Y) \quad &:= \quad \forall x (x \in X \rightarrow x \notin Y) \\
\text{and } \textsc{conn}(u, X, v) \quad &:= \quad \exists w (\textsc{adj}(u, w) \wedge w \in X) \wedge \exists w (\textsc{adj}(v, w) \wedge w \in X) \\
& \quad\; \wedge \forall A \forall B ((A \subseteq X \wedge B \subseteq X \wedge \textsc{dis}(A, B)) \rightarrow \exists a \exists b (a \in A \wedge b \in B \wedge \textsc{adj}(a, b)))
\end{aligned}
$$

The subformula $\textsc{conn}(u, X, v)$ expresses that u, v are adjacent to X and that $G[X]$ is connected, which implies that there exists a path from u to v in $G[X \cup \{u, v\}]$. By negation we can now express that G does not contain H as a topological minor, i.e., $G \models \neg\Phi_H$ iff G is H-topological-minor-free.