Faster Parameterized Algorithms for Modification Problems to Minor-Closed Classes

Laure Morelle
LIRMM, Université de Montpellier, CNRS, France

Ignasi Sau
LIRMM, Université de Montpellier, CNRS, France

Giannos Stamoulis
LIRMM, Université de Montpellier, CNRS, France

Dimitrios M. Thilikos
LIRMM, Université de Montpellier, CNRS, France

Abstract

Let $\mathcal{G}$ be a minor-closed graph class and let G be an n-vertex graph. We say that G is a *k-apex* of $\mathcal{G}$ if G contains a set S of at most k vertices such that $G \setminus S$ belongs to $\mathcal{G}$. Our first result is an algorithm that decides whether G is a k-apex of $\mathcal{G}$ in time $2^{\mathsf{poly}(k)} \cdot n^2$. This algorithm improves the previous one, given by Sau, Stamoulis, and Thilikos [ICALP 2020, TALG 2022], whose running time was $2^{\mathsf{poly}(k)} \cdot n^3$. The *elimination distance* of G to $\mathcal{G}$, denoted by $\mathsf{ed}_{\mathcal{G}}(G)$, is the minimum number of rounds required to reduce each connected component of G to a graph in $\mathcal{G}$ by removing one vertex from each connected component in each round. Bulian and Dawar [Algorithmica 2017] proved the existence of an FPT-algorithm, with parameter k, to decide whether $\mathsf{ed}_{\mathcal{G}}(G) \leq k$. This algorithm is based on the computability of the minor-obstructions and its dependence on k is not explicit. We extend the techniques used in the first algorithm to decide whether $\mathsf{ed}_{\mathcal{G}}(G) \leq k$ in time $2^{2^{2^{\mathsf{poly}(k)}}} \cdot n^2$. This is the first algorithm for this problem with an explicit parametric dependence in k. In the special case where $\mathcal{G}$ excludes some apex-graph as a minor, we give two alternative algorithms, one running in time $2^{2^{\mathcal{O}(k^2 \log k)}} \cdot n^2$ and one running in time $2^{\mathsf{poly}(k)} \cdot n^3$. As a stepping stone for these algorithms, we provide an algorithm that decides whether $\mathsf{ed}_{\mathcal{G}}(G) \leq k$ in time $2^{\mathcal{O}(\mathsf{tw} \cdot k + \mathsf{tw} \log \mathsf{tw})} \cdot n$, where tw is the treewidth of G. This algorithm combines the dynamic programming framework of Reidl, Rossmanith, Villaamil, and Sikdar [ICALP 2014] for the particular case where $\mathcal{G}$ contains only the empty graph (i.e., for treedepth) with the representative-based techniques introduced by Baste, Sau, and Thilikos [SODA 2020]. In all the algorithmic complexities above, poly is a polynomial function whose degree depends on $\mathcal{G}$, while the hidden constants also depend on $\mathcal{G}$. Finally, we provide explicit upper bounds on the size of the graphs in the minor-obstruction set of the class of graphs $\mathcal{E}_k(\mathcal{G}) = \{G \mid \mathsf{ed}_{\mathcal{G}}(G) \leq k\}$.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Graph minors, Parameterized algorithms, Graph modification problems, Vertex deletion, Elimination distance, Irrelevant vertex technique, Flat Wall Theorem, Obstructions

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.93

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://doi.org/10.48550/arXiv.2210.02167`

Funding Supported by the ANR projects ELIT (ANR-20-CE48-0008-01), ESIGMA (ANR-17-CE23-0010), and the French-German Collaboration ANR/DFG Project UTMA (ANR-20-CE92-0027).

Acknowledgements We would like to thank the reviewers for helpful remarks that improved the presentation of the article.

1 Introduction

The *distance from triviality* is a concept formalized by Guo, Hüffner, and Niedermeier [24] to express the closeness of a graph to a supposedly "simple" target graph class. One such a measure of closeness is, for instance, the number of vertices or edges that one must delete/add from/to a graph G to obtain a graph in the target graph class. This concept of distance to a graph class has recently gained the interest of the parameterized complexity community. The motivation is that, if a problem is tractable on a graph class $\mathcal{G}$, it is natural to study other classes of graphs according to their "distance to $\mathcal{G}$". In this paper, we focus on two such measures of distance from triviality: Given a target graph class $\mathcal{G}$, we consider the *vertex deletion distance* to $\mathcal{G}$ and the *elimination distance* to $\mathcal{G}$, which we formalize next.

Given a target graph class $\mathcal{G}$ and a non-negative integer k, we define $\mathcal{A}_k(\mathcal{G})$ as the set of all graphs containing a set S of at most k vertices whose removal results in a graph in $\mathcal{G}$. If $G \in \mathcal{A}_k(\mathcal{G})$, then we say that G is a *k-apex* of $\mathcal{G}$. We refer to S as a *k-apex set of G for the class $\mathcal{G}$*. In other words, we consider the following meta-problem for a fixed class $\mathcal{G}$.

VERTEX DELETION TO $\mathcal{G}$
Input: A graph G and a non-negative integer k.
Objective: Find, if it exists, a k-apex set of G for the class $\mathcal{G}$.

Throughout the paper, we denote by n (resp. m) the number of vertices (resp. edges) of the input graph of the problem under consideration. The importance of VERTEX DELETION TO $\mathcal{G}$ can be illustrated by the variety of graph modification problems that it encompasses. For instance, if $\mathcal{G}$ is the class of edgeless (resp. acyclic, planar, bipartite, (proper) interval, chordal) graphs, then we obtain the VERTEX COVER (resp. FEEDBACK VERTEX SET, VERTEX PLANARIZATION, ODD CYCLE TRANSVERSAL, (PROPER) INTERVAL VERTEX DELETION, CHORDAL VERTEX DELETION) problem.

The second measure of distance from triviality that we study was recently introduced by Bulian and Dawar [10, 11]. Given a graph class $\mathcal{G}$, we define the *elimination distance* of a graph G to $\mathcal{G}$, denoted by $\mathsf{ed}_{\mathcal{G}}(G)$, as follows:

$$\mathsf{ed}_{\mathcal{G}}(G) = \begin{cases} 0 & \text{if } G \in \mathcal{G}, \\ 1 + \min\{\mathsf{ed}_{\mathcal{G}}(G \setminus \{v\}) \mid v \in V(G)\} & \text{if } G \text{ is connected,} \\ \max\{\mathsf{ed}_{\mathcal{G}}(H) \mid H \text{ is a connected component of } G\} & \text{otherwise.} \end{cases}$$

Given that $\mathsf{ed}_{\mathcal{G}}(G) \leq k$, a set $S \subseteq V(G)$ of vertices recursively deleted from G to achieve $\mathsf{ed}_{\mathcal{G}}(G)$ is called a *k-elimination set of G for $\mathcal{G}$*. We define the (parameterized) class of graphs $\mathcal{E}_k(\mathcal{G}) = \{G \mid \mathsf{ed}_{\mathcal{G}}(G) \leq k\}$. The above notion can be seen as a natural generalization of *treedepth* (denoted by td), which corresponds to the case where $\mathcal{G}$ contains only the empty graph. Treedepth, along with treewidth, are two of the most studied and widely used parameters to measure the structural complexity of a graph [12, 36, 43]. The second meta-problem that we consider is the following, again for a fixed graph class $\mathcal{G}$.

ELIMINATION DISTANCE TO $\mathcal{G}$
Input: A graph G and a non-negative integer k.
Objective: Find, if it exists, a k-elimination set of G for the class $\mathcal{G}$.

Unsurprisingly, VERTEX DELETION TO $\mathcal{G}$ is NP-hard for every non-trivial graph class $\mathcal{G}$ [41], while ELIMINATION DISTANCE TO $\mathcal{G}$ is NP-hard even when $\mathcal{G}$ contains only the empty graph [45]. To circumvent this intractability, we study both problems from the parameterized complexity point of view and consider their parameterizations by k. In this setting, the most desirable behavior is the existence of an algorithm running in time $f(k) \cdot n^{\mathcal{O}(1)}$, where f is a computable function depending only on k. Such an algorithm is called *fixed-parameter tractable*, or FPT-*algorithm* for short, and a parameterized problem admitting an FPT-algorithm is said to belong to the parameterized complexity class FPT. Also, the function f is called *parametric dependence* of the corresponding FPT-algorithm, and the challenge is to design FPT-algorithms with small parametric dependencies and with a polynomial factor of small degree [12, 14, 17, 44]. We may also consider XP-*algorithms*, i.e., algorithms running in time $f(k) \cdot n^{g(k)}$ for some computable functions f and g depending only on k.

In general, for any of the two considered problems, we cannot expect FPT-algorithms for every graph class $\mathcal{G}$. For instance, the two problems are NP-hard, even for $k = 0$, for every graph class $\mathcal{G}$ whose recognition problem is NP-hard. This is the case of 3-colorable graphs, which is a class closed under taking (induced) subgraphs. In this paper, we focus on a family of graph classes that exhibits a nice behavior with respect to the considered problems (and many others): we consider $\mathcal{G}$ to be a *minor-closed* graph class, i.e., such that every minor of a graph in $\mathcal{G}$ (that is, obtained from a subgraph of a graph in $\mathcal{G}$ by contracting edges; see Section 2 for the formal definition) is also in $\mathcal{G}$. Indeed, it turns out that, for every such a family $\mathcal{G}$, the problems become fixed-parameter tractable, as we proceed to discuss.

The *minor-obstruction set* (in short *obstruction set*) of $\mathcal{G}$ is the set of minor-minimal graphs that do not belong to $\mathcal{G}$, and is denoted by $\mathsf{obs}(\mathcal{G})$. Notice that $\mathsf{obs}(\mathcal{G})$ gives a complete characterization of $\mathcal{G}$ as, for every graph G, it holds that $G \in \mathcal{G}$ if and only if, for every $H \in \mathsf{obs}(\mathcal{G})$, H is not a minor of G. Because of Robertson and Seymour's theorem [49], $\mathsf{obs}(\mathcal{G})$ is *finite* for every minor-closed graph class. As checking whether an h-vertex graph H is a minor of G can be done in time $f(h) \cdot n^2$ [31, 47], the finiteness of $\mathsf{obs}(\mathcal{G})$ along with the above characterization imply that, for every minor-closed graph class $\mathcal{G}$, checking whether $G \in \mathcal{G}$ can be done in time $c \cdot n^2$, where c is a constant depending on the graph class $\mathcal{G}$. This meta-theorem implies the *existence* of FPT-algorithms for a wide family of problems, including Vertex Deletion to $\mathcal{G}$ and Elimination Distance to $\mathcal{G}$. Indeed, this follows by observing that if $\mathcal{G}$ is minor-closed, then for every non-negative integer k, the classes $\mathcal{A}_k(\mathcal{G})$ and $\mathcal{E}_k(\mathcal{G})$ are also minor-closed.

As Robertson and Seymour's theorem [49] does *not* give any way to construct the corresponding obstruction sets, the aforementioned argument is not constructive, i.e., it cannot construct the obstruction sets required for the corresponding FPT-algorithms. Moreover, these algorithms are *non-uniform* in k, meaning that we have a *distinct* algorithm for every value of k. Important steps towards the constructibility of such FPT-algorithms were done by Adler, Grohe, and Kreutzer [1] and Bulian and Dawar [11], who respectively proved that $\mathsf{obs}(\mathcal{A}_k(\mathcal{G}))$ and $\mathsf{obs}(\mathcal{E}_k(\mathcal{G}))$ are effectively computable. Hence, for both problems, it is possible to construct uniform (in k) algorithms running in time $f(k) \cdot n^2$ for some computable function f. However, this does not imply any reasonable, or even explicit, parametric dependence of the obtained algorithms.

The main focus of this paper is on the parametric and polynomial dependence of FPT-algorithms to solve Vertex Deletion to $\mathcal{G}$ and Elimination Distance to $\mathcal{G}$, i.e., for recognizing the classes $\mathcal{A}_k(\mathcal{G})$ and $\mathcal{E}_k(\mathcal{G})$, when $\mathcal{G}$ is a minor-closed graph class.

Concerning Vertex Deletion to $\mathcal{G}$, after a number of articles for particular cases of minor-closed classes $\mathcal{G}$, such as graphs of bounded treewidth [19, 34], planar graphs [27, 42], or graphs of bounded genus [37], an explicit FPT-algorithm for *any* minor-closed graph $\mathcal{G}$ was recently proposed by Sau, Stamoulis, and Thilikos [51], running in time $2^{\mathcal{O}(k^c)} \cdot n^3$, where c is a constant that depends on the maximum size of a graph in the obstruction set of $\mathcal{G}$.

Moreover, in the case where $\mathsf{obs}(\mathcal{G})$ contains some apex-graph (that is, a 1-apex for the class of planar graphs), Sau, Stamoulis, and Thilikos [51] gave an improved running time of $2^{\mathcal{O}(k^c)} \cdot n^2$. Note also that the more general variant where $\mathcal{G}$ is a topological-minor-closed graph class is in FPT as well [20].

As for Elimination Distance to $\mathcal{G}$ when $\mathcal{G}$ is minor-closed, no explicit parametric dependence was known, with the notable exception of treedepth, for which Reidl, Rossmanith, Villaamil, and Sikdar [46] gave an algorithm deciding whether $\mathsf{td}(G) \leq k$ in time $2^{\mathcal{O}(k\cdot\mathsf{tw})} \cdot n$, where $\mathsf{tw} := \mathsf{tw}(G)$ (see also [9]). Using our terminology, and given that $\mathsf{tw}(G) \leq \mathsf{td}(G)$ for every graph G, this yields an FPT-algorithm for Elimination Distance to $\mathcal{G}_\emptyset$, where $\mathcal{G}_\emptyset$ is the class consisting of the empty graph, running in time $2^{\mathcal{O}(k^2)} \cdot n$. Note that this algorithm [46], combined with the fact that $\mathsf{td}(G) \leq \log(n) \cdot \mathsf{tw}(G)$ (see [9]), imply an XP-algorithm for the problem of computing td when parameterized by tw, namely an algorithm that computes the value of $\mathsf{td}(G)$ in time $n^{\mathcal{O}(\mathsf{tw}(G)^2)}$. To the best of our knowledge, it is open whether computing td parameterized by tw is in FPT.

Before describing our results, let us mention some recent relevant results dealing with Elimination Distance to $\mathcal{G}$ for classes $\mathcal{G}$ that are not necessarily minor-closed. Agrawal and Ramanujan [4] (resp. Agrawal, Kanesh, Panolan, Ramanujan, and Saurabh [3]) provided FPT-algorithms, with parameter k, when $\mathcal{G}$ is the class of cliques (resp. graphs of bounded degree). Fomin, Golovach, and Thilikos [18] identified sufficient and necessary conditions for the existence of FPT-algorithms when $\mathcal{G}$ is definable in first-order logic (such as having bounded degree). Jansen, de Kroon, and Włodarczyk [26] proved, among other results, that if $\mathcal{G}$ is a hereditary union-closed graph class and Vertex Deletion to $\mathcal{G}$ can be solved in time $2^{k^{\mathcal{O}(1)}} \cdot n^{\mathcal{O}(1)}$ (as it is the case for every minor-closed class $\mathcal{G}$ by the results of [51]), then there is an algorithm that, given an n-vertex graph G, computes an $\mathcal{O}(\mathsf{ed}_{\mathcal{G}}(G)^3)$-elimination set of G for $\mathcal{G}$ in time $2^{\mathsf{ed}_{\mathcal{G}}(G)^{\mathcal{O}(1)}} \cdot n^{\mathcal{O}(1)}$. Therefore, for union-closed minor-closed graph classes $\mathcal{G}$, the result of [26] yields an FPT-approximation algorithm for Elimination Distance to $\mathcal{G}$.

Agrawal, Kanesh, Lokshtanov, Panolan, Ramanujan, Saurabh, and Zehavi [2] proved that if $\mathcal{G}$ is hereditary, union-closed, and definable in monadic second-order logic, then Vertex Deletion to $\mathcal{G}$ is (non-uniformly) in FPT if, and only if, Elimination Distance to $\mathcal{G}$ is (non-uniformly) in FPT. Incidentally, they also showed that if $\mathcal{G}$ is defined by excluding a finite number of connected topological minors, then Elimination Distance to $\mathcal{G}$ is (uniformly) in FPT. We note that the results of [2] do not provide explicit parametric dependencies for these FPT-algorithms. Also, let us mention that it was conjectured in [2] that Elimination Distance to $\mathcal{G}$ is in FPT parameterized by a generalization of treewidth called *$\mathcal{G}$-treewidth* (see [2, 16, 26]). Note that, if true, this conjecture would answer the open problem mentioned above of whether computing td parameterized by tw is in FPT.

Our results. In this paper, we provide explicit FPT-algorithms for Vertex Deletion to $\mathcal{G}$ and Elimination Distance to $\mathcal{G}$ for *every* fixed minor-closed graph class $\mathcal{G}$. Our first result is the following.

▶ **Theorem 1.** *For every minor-closed graph class $\mathcal{G}$, there exists an algorithm that solves* Vertex Deletion to $\mathcal{G}$ *in time $2^{\mathsf{poly}(k)} \cdot n^2$.*

The degree of the polynomial function poly in the running time of Theorem 1 and of the other results below, as well as the constants hidden in the $\mathcal{O}$-notation in the running time of the algorithms, depend on the maximum size of a graph in $\mathsf{obs}(\mathcal{G})$. Thus, the algorithm of Theorem 1, while being uniformly FPT in k, is *not* uniform in the target class $\mathcal{G}$, as one needs to know an upper bound on the size of the minor-obstructions. This “meta-non-uniformity” applies to all the algorithms presented in this paper, and it is also the case, among many others, of the FPT-algorithms in [51]. The algorithm of Theorem 1 improves the algorithm of [51] from cubic to quadratic complexity in n while keeping the same parametric dependence on k. This answers positively one of the open problems posed in [51].

Our next algorithmic results concern Elimination Distance to $\mathcal{G}$ and provide, to the authors' knowledge, the first FPT-algorithms for this problem, when $\mathcal{G}$ is minor-closed, with an *explicit parametric dependence.*

▶ **Theorem 2.** *For every minor-closed graph class $\mathcal{G}$, there exists an algorithm that solves* Elimination Distance to $\mathcal{G}$ *in time* $2^{2^{2^{\mathsf{poly}(k)}}} \cdot n^2$. *In the particular case where* $\mathsf{obs}(\mathcal{G})$ *contains an apex-graph, this algorithm runs in time* $2^{2^{\mathcal{O}(k^2 \log k)}} \cdot n^2$.

Examples of classes $\mathcal{G}$ where $\mathsf{obs}(\mathcal{G})$ contains an apex-graph are graphs of bounded Euler genus, such as planar graphs. Our next result improves the parametric dependence of the algorithm of Theorem 2 when $\mathsf{obs}(\mathcal{G})$ contains an apex-graph, with a worse polynomial factor.

▶ **Theorem 3.** *For every minor-closed graph class $\mathcal{G}$ such that* $\mathsf{obs}(\mathcal{G})$ *contains an apex-graph, there exists an algorithm that solves* Elimination Distance to $\mathcal{G}$ *in time* $2^{\mathsf{poly}(k)} \cdot n^3$.

As discussed later, a crucial ingredient in the algorithms of Theorem 2 and Theorem 3 is to solve Elimination Distance to $\mathcal{G}$ parameterized by the treewidth of the input graph. The following result, which may be of independent interest, deals with this case.

▶ **Theorem 4.** *For every minor-closed graph class $\mathcal{G}$, there exists an algorithm that solves* Elimination Distance to $\mathcal{G}$ *in time* $2^{\mathcal{O}(k \cdot \mathsf{tw} + \mathsf{tw} \log \mathsf{tw})} \cdot n$, *where* tw *denotes the treewidth of the input graph.*

The algorithm of Theorem 4 can be seen as a generalization of the algorithm of Reidl, Rossmanith, Villaamil, and Sikdar [46] deciding whether $\mathsf{td}(G) \leq k$ in time $2^{\mathcal{O}(k \cdot \mathsf{tw})} \cdot n$. Since, for any graph G and any graph class $\mathcal{G}$, $\mathsf{ed}_{\mathcal{G}}(G) \leq \mathsf{td}(G) \leq \mathsf{tw}(G) \cdot \log n$, Theorem 4 implies the existence of an XP-algorithm for Elimination Distance to $\mathcal{G}$ parameterized by treewidth, when $\mathcal{G}$ is minor-closed, running in time $n^{\mathcal{O}(\mathsf{tw}^2)}$. Given that the conjecture of [2] is still open, this is the best type of algorithm that one can expect for Elimination Distance to $\mathcal{G}$ parameterized by treewidth.

Finally, for any minor-closed graph class $\mathcal{G}$, we provide an upper bound on the size of the graphs in the obstruction set of $\mathcal{E}_k(\mathcal{G})$.

▶ **Theorem 5.** *For every minor-closed graph class $\mathcal{G}$ and for every positive integer k, each graph in* $\mathsf{obs}(\mathcal{E}_k(\mathcal{G}))$ *has at most* $2^{2^{2^{2^{\mathsf{poly}(k)}}}}$ *vertices. Moreover, if* $\mathsf{obs}(\mathcal{G})$ *contains an apex-graph, this bound drops to* $2^{2^{\mathsf{poly}(k)}}$.

The only previously known bound for the graphs in $\mathsf{obs}(\mathcal{E}_k(\mathcal{G}))$ is the one for treedepth by Dvořák, Giannopoulou, and Thilikos [15], who proved that every graph in $\mathsf{obs}(\mathcal{E}_k(\mathcal{G}_\emptyset))$ has size at most $2^{2^{k-1}}$. Theorem 5 can be seen as a generalization of the results of Sau, Stamoulis, and Thilikos [52], who provided similar upper bounds for the graphs in $\mathsf{obs}(\mathcal{A}_k(\mathcal{G}))$.

These two results are, to the authors' knowledge, the first upper bounds on the size of the graphs in the obstruction set for the elimination distance parameter, and give, as an immediate consequence, the first known upper bound for the size of these obstruction sets.

Our techniques. This paper builds heavily on the techniques recently introduced in [51] in order to deal with Vertex Deletion to $\mathcal{G}$, which are based on exploiting the Flat Wall Theorem of Robertson and Seymour [47], namely the version proved by Kawarabayashi, Thomas, and Wollan [32] and its recent restatement by Sau, Stamoulis, and Thilikos [50]. In a nutshell, the idea of Theorem 1, Theorem 2, and Theorem 3 is that, as far as the treewidth of the input graph is sufficiently large as an appropriate function of k, it is possible to either "branch" into a number of subproblems that depends only on k and where the value of the parameter is strictly smaller, or to find an irrelevant vertex (i.e., a vertex that does not change the answer to the considered problem) and remove it from the graph. The irrelevant vertex technique originates from Robertson and Seymour [47] and is further developped in [50–52]. Once the treewidth is bounded, what remains is to apply the most efficient possible algorithm to solve the problem via dynamic programming on tree decompositions.

Let us focus more particularly on the techniques we use to prove Theorem 1. Contrary to the algorithm of [51] that solves Vertex Deletion to $\mathcal{G}$ for any minor-closed class $\mathcal{G}$, we *avoid using iterative compression*. This explains the improvement from cubic to quadratic complexity in n. The algorithm of Theorem 1 can be seen as an extension of the algorithm of [51] that solves Vertex Deletion to $\mathcal{G}$ in the particular case where $\mathsf{obs}(\mathcal{G})$ contains some apex-graph, and uses ideas that date back to the work of Marx and Schlotter [42] for the Planarization problem, that is, when $\mathcal{G}$ is the class of planar graphs. In Section 3 we provide a sketch of the algorithms claimed in Theorem 1, Theorem 2, and Theorem 3, and in Section 4 we present the algorithm of Theorem 1 in full detail, along with a proof of its correctness.

The proof of Theorem 4 consists of a dynamic programming algorithm that combines the framework of [46] for the particular case where $\mathcal{G}$ contains only the empty graph (i.e., for treedepth) with the representative-based techniques introduced in [5]. A bit more precisely, the idea is to encode the partial solutions (called *characteristic*) via sets of annotated trees with some additional properties. Here, the trees correspond to partial elimination trees and the annotations indicate the representatives, in the leaves of the elimination trees, with respect to the canonical equivalence relation defined for the target class $\mathcal{G}$. The size of the characteristic dominates the running time of the whole algorithm. As usual when dealing with dynamic programming, the formal description of the algorithm is quite technical and lengthy, and has been deferred to the full version of the paper.

Finally, to obtain the upper bound on the size of a graph $G \in \mathsf{obs}(\mathcal{E}_k(\mathcal{G}))$ claimed in Theorem 5, we proceed in two steps. First, we bound the treewidth of G by a function of k. To do so, we observe that if the treewidth of G is big enough, then there is a big enough wall in G, and we find an irrelevant vertex v for Elimination Distance to $\mathcal{G}$ in G. However, $G \setminus \{v\} \in \mathcal{E}_k(\mathcal{G})$ and $G \notin \mathcal{E}_k(\mathcal{G})$, hence we reach a contradiction. The second step is to bound the size of a minor-minimal obstruction of small treewidth. This uses the classic technique of Lagergren [39] (see also [21–23, 28, 29, 38, 40, 52]) combined with the encoding of the tables of the dynamic programming algorithm that we use to prove Theorem 4; see the full paper.

2 Preliminaries

In this section we give some basic definitions needed to understand the main body of the paper. Due to space limitations and the length of all the formal definitions, the complete preliminaries are provided in the full version of the paper (definitions and preliminary results regarding treedepth, treewidth and boundaried graphs, and framework of flat walls).

Minors and obstructions. A graph G' is a *minor* of a graph G, denoted by $G' \preceq G$, if G' can be obtained from G by a sequence of vertex removals, edge removals, and edge contractions. Let $\mathcal{G}$ be a graph class that is closed under taking minors. Recall that the *minor obstruction set* of $\mathcal{G}$ is defined as the set of all minor-minimal graphs that are not in $\mathcal{G}$, and is denoted by $\mathsf{obs}(\mathcal{G})$. Given a finite non-empty collection of non-empty graphs $\mathcal{F}$, we denote by $\mathsf{exc}(\mathcal{F})$ the set containing every graph G that excludes all graphs in $\mathcal{F}$ as minors. We call each graph in $\mathsf{exc}(\mathcal{F})$ *$\mathcal{F}$-minor-free*.

Restating the problems. Let $\mathcal{G}$ be a minor-closed graph class and $\mathcal{F}$ be its obstruction set. Clearly, Vertex Deletion to $\mathcal{G}$ is the same problem as asking, given a graph G and some $k \in \mathbb{N}$, for a vertex set $S \subseteq V(G)$ of at most k vertices such that $G \setminus S \in \mathsf{exc}(\mathcal{F})$. Following the terminology of [5–8, 19, 20, 34, 35, 51], we call this problem $\mathcal{F}$-M-Deletion. Likewise, Elimination Distance to $\mathcal{G}$ is the same problem as asking whether $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(G) \leq k$. We thus follow a similar notation and call this problem $\mathcal{F}$-M-Elimination Distance. Using the notation, $\{K_1\}$-M-Elimination Distance is the problem of asking whether $\mathsf{td}(G) \leq k$.

Some conventions. In the rest of the paper, we fix $\mathcal{G}$ to be a minor-closed graph class and $\mathcal{F}$ to be the set $\mathsf{obs}(\mathcal{G})$. From Robertson and Seymour's theorem [49], we know that $\mathcal{F}$ is a finite collection of graphs. Given a graph G, we define its *apex number* to be the smallest integer a for which there is a set $A \subseteq V(G)$ of size at most a such that $G \setminus A$ is planar. An *apex-graph* is a graph with apex number one. Also, we define the *detail* of G, denoted by $\mathsf{detail}(G)$, to be the maximum among $|E(G)|$ and $|V(G)|$. We define three constants depending on $\mathcal{F}$ that will be used throughout the paper whenever we consider such a graph family $\mathcal{F}$. We define $a_{\mathcal{F}}$ as the minimum apex number of a graph in $\mathcal{F}$, we set $s_{\mathcal{F}} := \max\{|V(H)| \mid H \in \mathcal{F}\}$, and we set $\ell_{\mathcal{F}} := \max\{\mathsf{detail}(H)| \mid H \in \mathcal{F}\}$. Given a tuple $\mathbf{t} = (x_1, \ldots, x_\ell) \in \mathbb{N}^\ell$ and two functions $\chi, \psi : \mathbb{N} \to \mathbb{N}$, we write $\chi(n) = \mathcal{O}_{\mathbf{t}}(\psi(n))$ in order to denote that there exists a computable function $\phi : \mathbb{N}^\ell \to \mathbb{N}$ such that $\chi(n) = \mathcal{O}(\phi(\mathbf{t}) \cdot \psi(n))$. Notice that $s_{\mathcal{F}} \leq \ell_{\mathcal{F}} \leq s_{\mathcal{F}}(s_{\mathcal{F}} - 1)/2$, and thus $\mathcal{O}_{\ell_{\mathcal{F}}}(\cdot) = \mathcal{O}_{s_{\mathcal{F}}}(\cdot)$. Observe also that $\mathcal{A}_k(\mathcal{G})$ and $\mathcal{E}_k(\mathcal{G})$ are $K_{s_{\mathcal{F}}+k}$-minor-free graph classes, and thus, due to [53], we can always assume that G has $\mathcal{O}_{s_{\mathcal{F}}}(k\sqrt{\log k} \cdot n)$ edges, otherwise we can directly conclude that (G, k) is a no-instance for both problems.

Walls and flat walls. In this paper we extensively deal with walls and flat walls, following the framework of [50]. Unfortunately, more than ten pages are required to provide all the technical notions to correctly present all this framework, that is necessary to use the tools developed in [50–52]. Thus, formal definitions are provided in the full version of the paper. More precisely, we introduce walls and several notions concerning them (just look at Figure 1 to understand what a wall is). We then provide the definitions of a rendition and a painting in order to define flat walls. There are a number of technical terms (such as tilts, influence, regular flatness pairs, ...) that are not the main focus of this work. Let us just mention that the perimeter of a flat wall of a graph G separates $V(G)$ into two sets X and Y with Y containing the wall. The *compass* of a flat wall is $G[Y]$.

We define canonical partitions and the notion of bidimensionality. Informally speaking, a *canonical partition* of a graph with respect to some wall W refers to a partition of the vertex set of a graph in bags that follow the structure of a wall subgraph of the given graph; see Figure 1 for an illustration. The *bidimensionality* of a vertex set X with respect to a wall W of a graph G intuitively expresses the "spread" of a set X in a W-canonical partition of G. The crucial idea is that a set X of small bidimensionality cannot "destroy" a large (flat) wall too much.

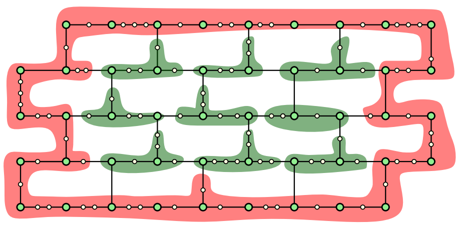

Figure 1 A 5-wall and its canonical partition $\mathcal{Q}$. The red bag is the external bag Q_{ext}.

Finally, we present homogeneous walls. Intuitively, *homogeneous flat walls* are flat walls that allow the routing of the same set of (topological) minors in the augmented flaps (i.e., the flaps together with the apex set) "cropped" by each one of their bricks. Such a homogeneous wall can be detected in a big enough flat wall (Proposition 10) and this "homogeneity" property implies that some central part of a big enough homogeneous wall can be declared irrelevant (Proposition 11).

3 Sketch of the algorithms

In this section we provide a sketch of the algorithms claimed in Theorem 1, Theorem 2 and Theorem 3. As mentioned in the introduction, Theorem 1 can be seen as a generalization of the algorithm of [51] that solves $\mathcal{F}$-M-Vertex Deletion in the particular case where $\mathcal{F}$ contains some apex-graph. While many techniques taken from [51] remain the same, some new ingredients are needed so as to deal with the possible existence of many apices in all graphs in $\mathcal{F}$. On the other hand, Theorem 2 and Theorem 3 can be seen as an adaptation of Theorem 1 to $\mathcal{F}$-M-Elimination Distance. Since these three algorithms follow a common streamline, we sketch all of them simultaneously while pointing out the steps where they differ. Moreover, the full proof of Theorem 1 is given in Section 4, while the proofs of Theorem 2 and Theorem 3 can be found in the full version of the paper.

The first common step is to run the following algorithm that states that a graph G in $\mathcal{A}_k(\mathsf{exc}(\mathcal{F}))$ or $\mathcal{E}_k(\mathsf{exc}(\mathcal{F}))$ either has bounded treewidth or contains a large wall. This result was proved in [51] in the case of $\mathcal{F}$-M-Deletion. The proof in the case of $\mathcal{F}$-M-Elimination Distance, necessary for Theorem 2 and Theorem 3, can be found in the full version of the paper.

▶ **Proposition 6** ([51], full paper). *Let $\mathcal{F}$ be a finite collection of graphs. There exist a function $f_1 : \mathbb{N} \to \mathbb{N}$ and an algorithm with the following specifications:*

`Find-Wall`(G, r, k)
***Input**: A graph G, an odd $r \in \mathbb{N}_{\geq 3}$, and $k \in \mathbb{N}$.*
***Output**: One of the following:*

- **Case 1:** *Either a report that (G, k) is a* no*-instance of $\mathcal{F}$-M-Deletion (resp. $\mathcal{F}$-M-Elimination Distance), or*
- **Case 2:** *a report that G has treewidth at most $f_1(s_{\mathcal{F}}) \cdot r + k$, or*
- **Case 3:** *an r-wall W of G.*

Moreover, $f_1(s_{\mathcal{F}}) = 2^{\mathcal{O}(s_{\mathcal{F}}^2 \cdot \log s_{\mathcal{F}})}$, and the algorithm runs in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(r^2+(k+r)\cdot\log(k+r))} \cdot n$ (resp. $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(r^2+k^2)} \cdot n$).

In **Case 1**, we can immediately conclude. In **Case 2**, since the treewidth of G is bounded, we use a dynamic programming algorithm to solve the corresponding problem. Namely, we solve $\mathcal{F}$-M-Deletion on instances of bounded treewidth using the main result from [5].

▶ **Proposition 7** ([5]). *For every finite collection of graphs $\mathcal{F}$, there exists an algorithm that, given a triple (G, tw, k) where G is a graph of treewidth at most* tw *and k is a non-negative integer, solves $\mathcal{F}$-*M-Deletion *in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(\mathsf{tw} \cdot \log \mathsf{tw})} \cdot n$.*

For $\mathcal{F}$-M-Elimination Distance, we use Theorem 4 to conclude. The proof of this (quite technically involved) dynamic programming algorithm is given in the full version of the paper.

Therefore, it only remains to deal with **Case 3**. Given an r-wall W of G, we want to reduce the size of G. To do so, we observe that we can either:

- **Case 3a:** find a subwall W_a of W and an apex set A_a such that W_a is flat in $G \setminus A_a$ and has a compass of bounded treewidth, or
- **Case 3b:** find a subwall W_b of W that is very "well connected" to an apex set A_b of small size.

The above distinction is done using two algorithmic versions of the Flat Wall Theorem consecutively. The first one comes from [32, Theorem 7.7] and is translated here in the new framework with tilts of [50]. Informally, we say that a graph H is *grasped* by a wall W in a graph G if there is a model of H in G such that the model of every node of H intersects W.

▶ **Proposition 8** ([32]). *There are two functions $f_2, f_3 : \mathbb{N} \to \mathbb{N}$, such that the images of f_2 are odd integers, and an algorithm with the following specifications:*

`Grasped-or-Flat`(G, r, t, W)
Input: *A graph G, an odd $r \in \mathbb{N}_{\geq 3}$, $t \in \mathbb{N}_{\geq 1}$, and an $f_2(t) \cdot r$-wall W of G.*
Output: *One of the following:*

- *Either a model of a K_t-minor in G grasped by W, or*
- *a set $A \subseteq V(G)$ of size at most $f_3(t)$ and a flatness pair $(W', \mathfrak{R}')$ of $G \setminus A$ of heigth r such that W' is a $\tilde{W}'$-tilt of some subwall $\tilde{W}'$ of W.*

Moreover, $f_2(t) = \mathcal{O}(t^{26})$, $f_3(t) = \mathcal{O}(t^{24})$, and the algorithm runs in time $\mathcal{O}(t^{24}m + n)$.

We would like to mention that the notion of being grasped by a wall is one of the new main arguments yielding the improvement of the complexity for $\mathcal{F}$-M-Deletion compared to [51].

The second one comes from [51] and adds the condition that W' has a compass of bounded treewidth, at the price of dropping the condition that the model of K_t is grasped by W.

▶ **Proposition 9** ([51]). *There exist a function $f_4 : \mathbb{N} \to \mathbb{N}$ and an algorithm with the following specifications:*

`Clique-Or-twFlat`(G, r, t)
Input: *A graph G, an odd $r \in \mathbb{N}_{\geq 3}$, and $t \in \mathbb{N}_{\geq 1}$.*
Output: *One of the following:*

- *Either a report that K_t is a minor of G, or*
- *a tree decomposition of G of width at most $f_4(t) \cdot r$, or*
- *a set $A \subseteq V(G)$ of size at most $f_3(t)$ and a regular flatness pair $(W', \mathfrak{R}')$ of $G \setminus A$ of height r whose $\mathfrak{R}'$-compass has treewidth at most $f_4(t) \cdot r$.*

Moreover, $f_4(t) = 2^{\mathcal{O}(t^2 \log t)}$ and this algorithm runs in time $2^{\mathcal{O}_t(r^2)} \cdot n$. The algorithm can be modified to obtain an explicit dependence on t in the running time, namely $2^{2^{\mathcal{O}(t^2 \log t)} \cdot r^3 \log r} \cdot n$.

`Grasped-or-Flat` is used to find a big enough complete graph "controlled" by the input wall, while we need `Clique-or-twFlat` to find a flat wall whose compass has bounded treewidth. Unfortunately, we cannot obtain both conditions simultaneously, and this is why we need both results. If, after using both algorithms, we obtain a flatness pair $(\tilde{W}', \mathfrak{R}')$ of

$G \setminus A_a$ of heigth r_a whose compass has bounded treewidth, then we are in **Case 3a**. In that case, the following result from [51] provides an algorithm that, given a flatness pair of big enough height, outputs a homogeneous flatness pair.

▶ **Proposition 10** ([51]). *There is a function $f_5 : \mathbb{N}^4 \to \mathbb{N}$, whose images are odd integers, and an algorithm with the following specifications:*

`Homogeneous`$(r, \tilde{a}, a, \ell, t, G, A, W, \mathcal{R})$
Input: *Five integers $r \in \mathbb{N}_{\geq 3}$, $\tilde{a}, a, \ell, t \in \mathbb{N}$, where $\tilde{a} \leq a$, a graph G, a set $A \subseteq V(G)$ of size at most a, and a flatness pair $(W, \mathfrak{R})$ of $G \setminus A$ of height $f_5(r, a, \tilde{a}, \ell)$ whose $\mathfrak{R}$-compass has treewidth at most t.*
Output: *A flatness pair $(\breve{W}, \breve{\mathfrak{R}})$ of $G \setminus A$ of height r that is ℓ-homogeneous with respect to $\binom{A}{\leq \tilde{a}}$ and is a W'-tilt of $(W, \mathfrak{R})$ for some subwall W' of W.*
Moreover, $f_5(r, \tilde{a}, a, \ell) = \mathcal{O}(r^{f_6(\tilde{a}, a, \ell)})$ where $f_6(\tilde{a}, a, \ell) = 2^{a^{\tilde{a}} \cdot 2^{\mathcal{O}((\tilde{a}+\ell)\cdot \log(\tilde{a}+\ell))}}$ and the algorithm runs in time $2^{\mathcal{O}(f_6(\tilde{a}, a, \ell) \cdot r \log r + t \log t)} \cdot (n + m)$.

Then we use the next result, that essentially says that the central vertex v of a big enough homogeneous wall is irrelevant, i.e., (G, k) and $(G \setminus v, k)$ are equivalent instances of the corresponding problem. Here, $\mathsf{bid}_{G \setminus A, W}(X)$ denotes the bidimensionality of a set X in the wall W with apex set A.

▶ **Proposition 11** ([52]). *Let $\mathcal{F}$ be a finite collection of graphs. There exist two functions $f_7 : \mathbb{N}^4 \to \mathbb{N}$ and $f_8 : \mathbb{N}^2 \to \mathbb{N}$, and an algorithm with the following specifications:*

`Find-Irrelevant-Vertex`$(k, a, G, A, W, \mathcal{R})$
Input: *Two integers $k, a \in \mathbb{N}$, a graph G, a set $A \subseteq V(G)$, and a regular flatness pair $(W, \mathcal{R})$ of $G \setminus A$ of height at least $f_7(a, \ell_{\mathcal{F}}, 3, k)$ that is $f_8(a, \ell_{\mathcal{F}})$-homogeneous with respect to $\binom{A}{\leq a}$.*
Output: *A vertex v of $G \setminus A$ such that for every set $X \subseteq V(G)$ with $\mathsf{bid}_{G \setminus A, W}(X) \leq k$ and $|A \setminus X| \leq a$, it holds that $G \setminus X \in \mathsf{exc}(\mathcal{F})$ if and only if $G \setminus (X \setminus v) \in \mathsf{exc}(\mathcal{F})$.*
Moreover, $f_7(a, \ell_{\mathcal{F}}, q, k) = \mathcal{O}(k \cdot (f_{\mathsf{ul}}(16a + 12\ell_{\mathcal{F}}))^3 + q)$, where f_{ul} is the function of the Unique Linkage Theorem (see [33]) and $f_8(a, \ell_{\mathcal{F}}) = a + \ell_{\mathcal{F}} + 3$, and this algorithm runs in time $\mathcal{O}(n + m)$.

We can prove that both k-apex sets and k-elimination sets have small bidimensionality. If, for every k-apex set S, $G \setminus S \in \mathsf{exc}(\mathcal{F})$ if and only if $G \setminus (S \setminus v) \in \mathsf{exc}(\mathcal{F})$, then it is straightforward to see that v is irrelevant for $\mathcal{F}$-M-Deletion. It is slightly less trivial to prove that, for each k-elimination set S, we can find some superset $X \supseteq S$ of small bidimensionality such that a similar statement holds. Additional details are available in the full paper.

Therefore we can recursively solve the problems on the instance $(G \setminus v, k)$.

If no flatness pair whose compass has bounded treewidth was found, then we are in **Case 3b**. In this case, inspired by [42] and [51], we use the following result of [52] that basically says that if there is a big enough flat wall W and an apex set A' of $a_{\mathcal{F}}$ vertices that are all adjacent to many bags of a canonical partition of W, then each k-apex set or k-elimination set intersects A'.

▶ **Proposition 12** ([52]). *There exist three functions $f_9, f_{10}, f_{11} : \mathbb{N}^3 \to \mathbb{N}$, such that if G is a graph, $k \in \mathbb{N}$, A is a subset of $V(G)$, $(W, \mathfrak{R})$ is a flatness pair of $G \setminus A$ of height at least $f_9(a_{\mathcal{F}}, s_{\mathcal{F}}, k)$, $\tilde{\mathcal{Q}}$ is a W-canonical partition of $G \setminus A$, A' is a subset of vertices of A that are adjacent, in G, to vertices of at least $f_{10}(a_{\mathcal{F}}, s_{\mathcal{F}}, k)$ many $f_{11}(a_{\mathcal{F}}, s_{\mathcal{F}}, k)$-internal bags of $\tilde{\mathcal{Q}}$, and $|A'| \geq a_{\mathcal{F}}$, then for every set $X \subseteq V(G)$ such that $G \setminus X \in \mathsf{exc}(\mathcal{F})$ and $\mathsf{bid}_{G \setminus A, W}(X) \leq k$, it holds that $X \cap A' \neq \emptyset$. Moreover, $f_9(a, s, k) = \mathcal{O}(2^a \cdot s^{5/2} \cdot k^{5/2})$, $f_{10}(a, s, k) = \mathcal{O}(2^a \cdot s^3 \cdot k^3)$, and $f_{11}(a, s, k) = \mathcal{O}((a^2 + k) \cdot s)$, where $a = a_{\mathcal{F}}$ and $s = s_{\mathcal{F}}$.*

For the $\mathcal{F}$-M-Deletion problem, if we find such a set A', then we can branch by guessing which vertex $v \in A'$ belongs to a k-apex set and recursively solving $(G \setminus v, k-1)$. Given that A' has size $a_{\mathcal{F}}$ and that k decreases after each guess, this step is applied at most $a_{\mathcal{F}}^k$ times.

However, for $\mathcal{F}$-M-Elimination Distance, k does not decrease, given that the size of a k-elimination set may not depend on k. Thus, this step may be done $a_{\mathcal{F}}^n$ times, which does not give an FPT-algorithm. To circumvent this problem, we propose two alternatives:

Option 1: The first alternative is to only use **Case 3a**. This is possible given that $(K_{s_{\mathcal{F}}+k}, k)$ is a no-instance of both problems. Thus, when using the algorithms `Grasped-or-Flat` and `Clique-or-twFlat`, we force the outcome to be an apex set A and a flatness pair of $G \setminus A$. However, the bound on the size of A now depends on k, and thus, so does the variable a in the input of the algorithm `Homogeneous`. This explains the triple-exponential parametric dependence on k in Theorem 2. Interestingly, a precise analysis of the time complexity shows that if $a_{\mathcal{F}} = 1$, i.e., when $\mathcal{F}$ contains an apex graph, the parametric dependence is only double-exponential on k (cf. Theorem 2).

Option 2: The second alternative is to restrict ourselves to the case where $a_{\mathcal{F}} = 1$. Thus, in **Case 3b**, we find a vertex v that belongs to every k-elimination set. There is no need to branch, and this step is done at most n times. However, the fact that the time complexity of this step is quadratic in n explains the cubic complexity of the algorithm in Theorem 3.

It remains to show that if no flatness pair whose compass has bounded treewidth was found, then we can find a flatness pair and a set A' satisfying the conditions of Proposition 12. To do so, using flow techniques, we find the set A of vertices with sufficiently many internally-disjoint paths to W, independently from one another. If this set is too large, we can safely declare a no-instance. Otherwise, we extend the canonical partition of W and just check whether $a_{\mathcal{F}}$ vertices of A are adjacent to many vertices of this new canonical partition. If this happens, then we can safely use Proposition 12. The second main improvement with respect to the algorithm in [51] is the new argument that the extension of the canonical partition of W can be done in a totally arbitrary manner. The quadratic complexity of this step stems from the search for internally-disjoint paths for every vertex of the input graph.

4 Vertex deletion to a minor-closed graph class

In this section we prove our main result for the $\mathcal{F}$-M-Deletion problem, namely Theorem 1.

All the propositions necessary for this proof have already been stated in Section 3, aside from the following result proved in [52] that intuitively states that, given a canonical partition $\tilde{\mathcal{Q}}$ of a flat wall $(W, \mathfrak{R})$ of big enough height, we can find a "packing" of subwalls of W that are inside some central part of W and such that the vertex set of every bag of $\tilde{\mathcal{Q}}$ intersects the vertices of at most one of these walls.

▶ **Proposition 13** ([52]). *There exists a function $f_{12} : \mathbb{N}^3 \to \mathbb{N}$ such that if $p, l \in \mathbb{N}_{\geq 1}$, $r \in \mathbb{N}_{\geq 3}$ is an odd integer, G is a graph, $(W, \mathfrak{R})$ is a flatness pair of G of height at least $f_{12}(l, r, p)$, and $\tilde{\mathcal{Q}}$ is a W-canonical partition of G, then there is a collection $\mathcal{W} = \{W^1, \ldots, W^l\}$ of r-subwalls of W such that*

- *for every $i \in [l]$, $\bigcup \mathsf{influence}_{\mathfrak{R}}(W^i)$ is a subgraph of $\bigcup\{Q \mid Q$ is a p-internal bag of $\tilde{\mathcal{Q}}\}$ and*
- *for every $i, j \in [l]$, with $i \neq j$, there is no internal bag of $\tilde{\mathcal{Q}}$ that contains vertices of both $V(\bigcup \mathsf{influence}_{\mathfrak{R}}(W^i))$ and $V(\bigcup \mathsf{influence}_{\mathfrak{R}}(W^j))$.*

Moreover, $f_{12}(l, r, p) = \mathcal{O}(\sqrt{l} \cdot r + p)$ and $\mathcal{W}$ can be constructed in time $\mathcal{O}(n + m)$.

4.1 Description of the algorithm for $\mathcal{F}$-M-Deletion

Our algorithm for $\mathcal{F}$-M-Deletion has three steps. In Step 1, either we can easily conclude with a positive or a negative answer (**Cases 1 and 2**) or we find a big wall. If we can find a large flat wall of bounded treewidth inside this wall, then we go to Step 2 and find an irrelevant vertex (**Case 3a**). Otherwise, we proceed to Step 3 where, by using flow techniques, we find a set of vertices that intersects every solution, and we branch on this set or we report a negative answer (**Case 3b**). The correctness of the algorithm is not trivial and will be justified in Subsection 4.2.

Given a non-negative integer x, we denote by $\mathsf{odd}(x)$ the smallest odd number that is not smaller than x. We define the following constants.

$$\begin{aligned}
a &= f_3(s_{\mathcal{F}} + a_{\mathcal{F}} - 1), & b &= f_3(s_{\mathcal{F}}),\\
q &= f_{10}(a_{\mathcal{F}}, s_{\mathcal{F}}, k), & p &= f_{11}(a_{\mathcal{F}}, s_{\mathcal{F}}, k),\\
l &= (q-1)\cdot(k+b), & r_6 &= f_7(a+b, \ell_{\mathcal{F}}, 3, k)\\
d &= f_8(a+b, \ell_{\mathcal{F}}) & r_5 &= f_5(r_6, a+b, a+b, d),\\
t &= f_4(s_{\mathcal{F}})\cdot r_5, & r_4 &= \mathsf{odd}(t+3),\\
r_3 &= f_{12}(a_{\mathcal{F}}, r_4, 1), & r_2 &= \mathsf{odd}(2 + f_2(s_{\mathcal{F}} + a_{\mathcal{F}} - 1)\cdot r_3),\\
r_2' &= \mathsf{odd}(\max\{f_9(a_{\mathcal{F}}, s_{\mathcal{F}}, k), f_{12}(l+1, r_2, p)\}), & r_1 &= \mathsf{odd}(f_2(s_{\mathcal{F}})\cdot r_2' + k).
\end{aligned}$$

Note that $r_6 = \mathcal{O}_{\ell_{\mathcal{F}}}(k)$, $r_5, r_4, r_3, r_2, t = \mathcal{O}_{\ell_{\mathcal{F}}}(k^c)$, and $r_2', r_1 = \mathcal{O}_{\ell_{\mathcal{F}}}(k^{c+2})$ where $c = f_6(a+b, a+b, d)$. Recall from Section 2 that we may assume that G has $\mathcal{O}_{s_{\mathcal{F}}}(k\sqrt{\log k}\cdot n)$ edges.

Step 1. Run the algorithm `Find-Wall` from Proposition 6 with input (G, r_1, k) and, in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(r_1^2 + (k+r_1)\log(k+r_1))}\cdot n = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{2(c+2)})}\cdot n$,

- either report a no-instance, or
- conclude that $\mathsf{tw}(G) \leq f_1(s_{\mathcal{F}})\cdot r_1 + k$ and solve $\mathcal{F}$-M-Deletion in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}((r_1+k)\log(r_1+k))}\cdot n = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{c+2}\cdot\log k)}\cdot n$ using the algorithm of Proposition 7, or
- obtain an r_1-wall W_1 of G.

If the output of Proposition 6 is an r_1-wall W_1, consider all the $\binom{r_1}{r_2}^2 = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^c\log k)}$ r_2-subwalls of W_1. For each one of them, say W_2, let W_2^* be the central (r_2-2)-subwall of W_2 and let D_{W_2} be the graph obtained from G after removing the perimeter of W_2 and taking the connected component containing W_2^*. Run the algorithm `Grasped-or-Flat` of Proposition 8 with input $(D_{W_2}, r_3, s_{\mathcal{F}} + a_{\mathcal{F}} - 1, W_2^*)$. This can be done in time $\mathcal{O}_{s_{\mathcal{F}}}(k\sqrt{\log k}\cdot n)$.

If for some of these subwalls the result is a set $A \subseteq V(D_{W_2})$ with $|A| \leq a$ and a flatness pair $(W_3, \mathfrak{R}_3)$ of $D_{W_2}\setminus A$ of height r_3 then, as in Proposition 13, compute a W_3-canonical partition $\tilde{\mathcal{Q}}$ of $D_{W_2}\setminus A$ and a collection $\mathcal{W} = \{W^1, \dots, W^{a_{\mathcal{F}}}\}$ of r_4-subwalls of W_3 such that for every $i \in [a_{\mathcal{F}}]$, $\bigcup \mathsf{influence}_{\mathfrak{R}_3}(W^i)$ is a subgraph of $\bigcup\{Q \mid Q \text{ is a } p\text{-internal bag of } \tilde{\mathcal{Q}}\}$ and for every $i, j \in [a_{\mathcal{F}}]$, with $i \neq j$, there is no internal bag of $\tilde{\mathcal{Q}}$ that contains vertices of both $V(\bigcup \mathsf{influence}_{\mathfrak{R}_3}(W^i))$ and $V(\bigcup \mathsf{influence}_{\mathfrak{R}_3}(W^j))$. This can be done in time $\mathcal{O}_{s_{\mathcal{F}}}(k\sqrt{\log k}\cdot n)$.

For $i \in [a_{\mathcal{F}}]$, let W^{i*} be the central (r_4-2)-subwall of W^i and let D_{W^i} be the graph obtained from D_{W_2} after removing A and the perimeter of W^i and taking the connected component containing W^{i*}. Run the algorithm `Clique-or-twFlat` of Proposition 9 with input $(D_{W^i}, r_5, s_{\mathcal{F}})$. This takes time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(r_5^2)}\cdot n = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{2c})}\cdot n$. If for one of these subwalls the result is a set A' of size at most b and a regular flatness pair $(W_5, \mathfrak{R}_5)$ of $D_{W^i}\setminus A'$ of height r_5 whose $\mathfrak{R}_5$-compass has treewidth at most t, then we proceed to Step 2.

If, for every flatness pair $(W_3, \mathfrak{R}_3)$ and for every $i \in [a_{\mathcal{F}}]$, the result is a report that $K_{s_{\mathcal{F}}}$ is a minor of D_{W^i}, then we proceed to Step 3.

Step 2 (irrelevant vertex case). We obtain a 7-tuple $\mathfrak{R}'_5$ by adding all vertices of $G \setminus V(\mathsf{Compass}_{\mathfrak{R}_5}(W_5))$ to the set in the first coordinate of $\mathfrak{R}_5$, such that $(W_5, \mathfrak{R}'_5)$ is a regular flatness pair of $G\setminus(A\cup A')$ whose $\mathfrak{R}'_5$-compass has treewidth at most t. We apply the algorithm `Homogeneous` of Proposition 10 with input $(r_6, a+b, a+b, d, t, G, A \cup A', W_5, \mathfrak{R}'_5)$, which outputs, in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(t \log t + k \log k)} \cdot n = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^c \log k)} \cdot n$, a flatness pair $(W_6, \mathfrak{R}_6)$ of $G\setminus(A\cup A')$ of height r_6 that is d-homogeneous with respect to $2^{A\cup A'}$ and is a W^*-tilt of $(W_5, \mathfrak{R}'_5)$ for some subwall W^* of W_5. We apply the algorithm `Find-Irrelevant-Vertex` of Proposition 11 with input $(k, a+b, G, A \cup A', W_6, \mathfrak{R}_6)$, which outputs, in time $\mathcal{O}(n+m) = \mathcal{O}_{\ell_{\mathcal{F}}}(k\sqrt{\log k} \cdot n)$, a vertex v such that (G, k) and $(G \setminus v, k)$ are equivalent instances of $\mathcal{F}$-M-Deletion. Then the algorithm runs recursively on the equivalent instance $(G \setminus v, k)$.

Step 3 (branching case). Consider all the r'_2-subwalls of W_1, which are at most $\binom{r_1}{r'_2}^2 = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{c+2} \log k)}$ many, and for each of them, say W'_2, compute its canonical partition $\mathcal{Q}$. Then, contract each bag Q of $\mathcal{Q}$ to a single vertex v_Q, and add a new vertex v_{all} and make it adjacent to all v_Q's. In the resulting graph G', for every vertex y of $G \setminus V(W'_2)$, check, using a path augmentation algorithm [13], whether there are q internally vertex-disjoint paths from v_{all} to y in time $\mathcal{O}(q \cdot m) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^4\sqrt{\log k} \cdot n)$. Let $\tilde{A}$ be the set of all such y's.

If $|\tilde{A}| < a_{\mathcal{F}}$, then report a no-instance.

If $a_{\mathcal{F}} \leq |\tilde{A}| \leq k+b$, then consider all the $\binom{|\tilde{A}|}{a_{\mathcal{F}}} = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(\log k)}$ subsets of $\tilde{A}$ of size $a_{\mathcal{F}}$. For each one of them, say A^*, construct $\tilde{\mathcal{Q}}$ by enhancing $\mathcal{Q}$ on $G \setminus A^*$. Then, we distinguish two cases depending on whether for every A^* all its vertices are adjacent to vertices of q p-internal bags of $\tilde{\mathcal{Q}}$.

If each vertex of A^* is adjacent to vertices of q p-internal bags of $\tilde{\mathcal{Q}}$, then (due to Proposition 12) A^* should intersect every solution of $\mathcal{F}$-M-Deletion for the instance (G, k). Therefore, the algorithm runs recursively on each instance $(G \setminus y, k-1)$ for $y \in A^*$. If one of them is a yes-instance with $(k-1)$-apex set S of $G \setminus y$, then (G, k) is a yes-instance with k-apex set $S \cup \{y\}$ of G. If all of them are no-instances, then report a no-instance. This concludes the case where each vertex of A^* is adjacent to vertices of q p-internal bags of $\tilde{\mathcal{Q}}$.

If for every subset A^* of $\tilde{A}$ of size $a_{\mathcal{F}}$, there is a vertex of A^* that is not adjacent to vertices of q p-internal bags of the given $\tilde{\mathcal{Q}}$, then report a no-instance. This concludes the case that $a_{\mathcal{F}} \leq |\tilde{A}| \leq k+b$.

If for every wall, $|\tilde{A}| > k+b$, then report that (G, k) is a no-instance of $\mathcal{F}$-M-Deletion.

Notice that Step 3, when applied, takes time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{c+2} \log k)} \cdot n^2$, because we apply the flow algorithms to each of the $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{c+2} \log k)}$ r'_2-subwalls and for each vertex of G. However, the search tree created by the branching algorithm has at most $a_{\mathcal{F}}$ branches and depth at most k. So Step 3 cannot be applied more than ${a_{\mathcal{F}}}^k$ times during the course of the algorithm. Since Step 1 runs in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{2(c+2)})} \cdot n$, Step 2 runs in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{2c})} \cdot n$, and both may be applied at most n times, the claimed time complexity follows: the algorithm runs in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{2(c+2)})} \cdot n^2$.

4.2 Correctness of the algorithm

Suppose first that (G, k) is a yes-instance and let S be a k-apex set of G. The application of the algorithm `Find-Wall` of Proposition 6 with input (G, r_1, k) either returns a report that $\mathsf{tw}(G) \leq f_1(s_{\mathcal{F}}) \cdot r_1 + k$ or returns an r_1-wall. In the first case, i.e., if $\mathsf{tw}(G) \leq f_1(s_{\mathcal{F}}) \cdot r_1 + k$, the application of the algorithm of Proposition 7 correctly outputs a k-apex set of G. We will focus on the latter case, i.e., where the algorithm `Find-Wall` returns an r_1-wall of G, say W_1. Since $r_1 \geq f_2(s_{\mathcal{F}}) \cdot r'_2 + k$, there is an $(f_2(s_{\mathcal{F}}) \cdot r'_2)$-subwall of W_1, say W_1^*, that does not contain vertices of S. Since $G \setminus S$ does not contain $K_{s_{\mathcal{F}}}$ as a minor, there is no model of $K_{s_{\mathcal{F}}}$ grasped by W_1^* and therefore, due to Proposition 8 with input $(G \setminus S, r'_2, s_{\mathcal{F}}, W_1^*)$, we know that there is a set $B \subseteq V(G \setminus S)$, with $|B| \leq b$, and a flatness pair $(W'_2, \mathfrak{R}'_2)$ of $G \setminus (S \cup B)$ of height r'_2 such that W'_2 is a W''-tilt of some subwall W'' of W_1^*.

Let $\mathcal{Q}$ be the canonical partition of W'_2. Let G' be the graph obtained by contracting each bag Q of $\mathcal{Q}$ to a single vertex v_Q, and adding a new vertex v_{all} and making it adjacent to all v_Q's. Let $\tilde{A}$ be the set of vertices y of $G \setminus V(W'_2)$ such that there are q internally vertex-disjoint paths from v_{all} to y in G'. We claim that $\tilde{A} \subseteq S \cup B$. To show this, we first prove that, for every $y \notin S \cup B$, the maximum number of internally vertex-disjoint paths from v_{all} to y in G' is $k + b + 4$. Indeed, if y is a vertex in the $\mathfrak{R}'_2$-compass of W'_2, there are at most $k + b$ such paths that intersect the set $S \cup B$ and at most four paths that do not intersect $S \cup B$ (in the graph $G' \setminus (S \cup B)$) due to the fact that $(W'_2, \mathfrak{R}'_2)$ is a flatness pair of $G \setminus (S \cup B)$. If y is not a vertex in the $\mathfrak{R}'_2$-compass of W'_2, then, since by the definition of flatness pairs the perimeter of W'_2 together with the set $S \cup B$ separate y from the $\mathfrak{R}'_2$-compass of W'_2, every collection of internally vertex-disjoint paths from v_{all} to y in G' should intersect the set $\{v_{Q_{\text{ext}}}\} \cup S \cup B$, where Q_{ext} is the external bag of $\mathcal{Q}$. Therefore, in both cases, if $y \notin S \cup B$, the maximum number of internally vertex-disjoint paths from v_{all} to y in G' is $k + b + 4$. Since $k + b + 4 < q$, we have that $y \notin \tilde{A}$. Hence, $\tilde{A} \subseteq S \cup B$ and therefore $|\tilde{A}| \leq k + b$. Hence, if (G, k) is a yes-instance we cannot have that $|\tilde{A}| > k + b$, so the algorithm correctly reports a no-instance at the end of Step 3.

Let $\tilde{\mathcal{Q}}$ be a W'_2-canonical partition of $G \setminus (S \cup B)$ obtained by enhancing $\mathcal{Q}$ on $G \setminus (S \cup B)$. Let $\tilde{A}'$ be the set of vertices in $S \cup B$ that are adjacent to vertices of at least q p-internal bags of $\tilde{\mathcal{Q}}$ (recall that $\tilde{A}$ is the set of vertices in $S \cup B$ that are adjacent to vertices of at least q internal bags of $\tilde{\mathcal{Q}}$). Note that $\tilde{A}' \subseteq \tilde{A}$ and therefore $|\tilde{A}'| \leq |\tilde{A}|$.

If $|\tilde{A}'| < a_{\mathcal{F}}$, then at most $a_{\mathcal{F}} - 1$ vertices of $S \cup B$ are adjacent to vertices of at least q p-internal bags of $\tilde{\mathcal{Q}}$. This means that the p-internal bags of $\tilde{\mathcal{Q}}$ that contain vertices adjacent to some vertex of $(S \cup B) \setminus \tilde{A}'$ are at most $(q - 1) \cdot (k + b) = l$.

Consider a family $\mathcal{W} = \{W^1, \ldots, W^{l+1}\}$ of $l + 1$ r_2-subwalls of W'_2 such that for every $i \in [l + 1]$, $\bigcup \mathsf{influence}_{\mathfrak{R}'_2}(W^i)$ is a subgraph of $\bigcup\{Q \mid Q \text{ is a } p\text{-internal bag of } \tilde{\mathcal{Q}}\}$ and for every $i, j \in [l + 1]$, with $i \neq j$, there is no internal bag of $\tilde{\mathcal{Q}}$ that contains vertices of both $V(\bigcup \mathsf{influence}_{\mathfrak{R}'_2}(W^i))$ and $V(\bigcup \mathsf{influence}_{\mathfrak{R}'_2}(W^j))$. The existence of $\mathcal{W}$ follows from Proposition 13 and the fact that $r'_2 \geq f_{12}(l + 1, r_2, p)$.

The fact that the p-internal bags of $\tilde{\mathcal{Q}}$ that contain vertices adjacent to some vertex of $(S \cup B) \setminus \tilde{A}'$ are at most l implies that there exists an $i \in [l + 1]$ such that no vertex of $V(\bigcup \mathsf{influence}_{\mathfrak{R}'_2}(W^i))$ is adjacent, in G, to a vertex in $(S \cup B) \setminus \tilde{A}'$. Let $W_2 := W^i$, let W_2^* be the central $(r_2 - 2)$-subwall of W_2, and let D_{W_2} be the graph obtained from G by removing the perimeter of W_2 and taking the connected component that contains W_2^*. Since no vertex of $V(\bigcup \mathsf{influence}_{\mathfrak{R}'_2}(W^i))$ is adjacent, in G, to a vertex in $(S \cup B) \setminus \tilde{A}'$, any path in D_{W_2} going from a vertex of W_2^* to a vertex in S must intersect a vertex of $\tilde{A}'$. Thus, there is no model of $K_{s_{\mathcal{F}} + a_{\mathcal{F}} - 1}$ grasped by W_2^* in D_{W_2}, because otherwise $K_{s_{\mathcal{F}}}$ would be a minor of $G \setminus S$. So, by applying the algorithm `Grasped-or-Flat` of Proposition 8 with input $(D_{W_2}, r_3, s_{\mathcal{F}} + a_{\mathcal{F}} - 1, W_2^*)$, since $r_2 - 2 \geq f_2(s_{\mathcal{F}} + a_{\mathcal{F}} - 1) \cdot r_3$, we should find a set $A \subseteq V(D_{W_2})$ with $|A| \leq a$ and a flatness pair $(W_3, \mathfrak{R}_3)$ of $D_{W_2} \setminus A$ of height r_3, such that W_3 is a tilt of some subwall $\tilde{W}_3$ of W_2.

Let $\tilde{\mathcal{Q}}'$ be a W_3-canonical partition of $D_{W_2} \setminus A$. Let $\mathcal{W}' = \{W^1, ..., W^{a_{\mathcal{F}}}\}$ be a collection of r_4-subwalls of W_3 such that for every $i \in [a_{\mathcal{F}}]$, $\bigcup \mathsf{influence}_{\mathfrak{R}_3}(W^i)$ is a subgraph of $\bigcup\{Q \mid Q$ is an internal bag of $\tilde{\mathcal{Q}}'\}$ and for every $i, j \in [a_{\mathcal{F}}]$, with $i \neq j$, there is no internal bag of $\tilde{\mathcal{Q}}'$ that contains vertices of both $V(\bigcup \mathsf{influence}_{\mathfrak{R}_3}(W^i))$ and $V(\bigcup \mathsf{influence}_{\mathfrak{R}_3}(W^j))$. Since $|\tilde{A}'| < a_{\mathcal{F}}$, there is an $i \in [a_{\mathcal{F}}]$ such that $V(\bigcup \mathsf{influence}_{\mathfrak{R}_3}(W^i))$ does not intersect $\tilde{A}'$. The existence of $\mathcal{W}'$ follows from Proposition 13 and the fact that $r_3 \geq f_{12}(a_{\mathcal{F}}, r_4, 1)$.

Let $W_4 := W^i$. Let W_4^* be the central $(r_4 - 2)$-subwall of W_4 and let D_{W_4} be the graph obtained from D_{W_2} after removing A and the perimeter of W_4 and taking the connected component containing W_4^*. Observe that any path between a vertex of S and a vertex of $V(\bigcup \mathsf{influence}_{\mathfrak{R}_3}(W_4))$ in D_{W_2} intersects $\tilde{A}'$. Since $\tilde{A}'$ does not intersect $V(\bigcup \mathsf{influence}_{\mathfrak{R}_3}(W_4))$, it implies that $\tilde{A}'$ does not intersect D_{W_4}, and thus $S \cap D_{W_4} = \emptyset$. Therefore, D_{W_4} is a minor of $G \setminus S$ and $K_{s_{\mathcal{F}}}$ is not a minor of D_{W_4}. Moreover, W_4^* is a wall of D_{W_4} of height $r_4 - 2 \geq t + 1$, so $\mathsf{tw}(D_{W_4}) > t = f_4(s_{\mathcal{F}}) \cdot r_5$. Therefore, by applying the algorithm `Clique-or-twFlat` of Proposition 9 with input $(D_{W_4}, r_5, s_{\mathcal{F}})$, we should obtain a set A' of size at most b and a regular flatness pair $(W_5, \mathfrak{R}_5)$ of $D_{W_4} \setminus A'$ of height r_5 whose $\mathfrak{R}_5$-compass has treewidth at most t. All this is checked in Step 1, and thus, the algorithm should run Step 2.

If $|\tilde{A}'| \geq a_{\mathcal{F}}$, then, due to Proposition 12 and the fact that $r_2' \geq f_9(a_{\mathcal{F}}, s_{\mathcal{F}}, k)$, for any set $X \subseteq V(G)$ such that $\mathsf{bid}_{G\setminus(S\cup B),W_2'}(X) \leq k$ and such that $G \setminus X \in \mathsf{exc}(\mathcal{F})$, it holds that $X \cap \tilde{A}' \neq \emptyset$. In particular, for any k-apex set S', $\mathsf{bid}_{G\setminus(S\cup B),W_2'}(S') \leq |S'| \leq k$, and thus $S' \cap \tilde{A}' \neq \emptyset$. Thus, there is a vertex $y \in \tilde{A}'$ such that $(G \setminus y, k-1)$ is a yes-instance. Hence, if the algorithm runs Step 3, it finds a vertex $y \in \tilde{A}'$ such that $(G \setminus y, k-1)$ is a yes-instance.

Note that the enhancement $\tilde{\mathcal{Q}}$ of the canonical partition $\mathcal{Q}$ is not unique. In particular, $\tilde{A}'$ depends on $\tilde{\mathcal{Q}}$. However, as long as there is such a $\tilde{\mathcal{Q}}$ such that $|\tilde{A}'| < a_{\mathcal{F}}$, the algorithm finds the wanted flatness pair $(W_4, \mathfrak{R}_4)$ in Step 1 and then runs Step 2. Hence, if (G, k) is a yes-instance, the algorithm runs Step 3 only if for all such $\tilde{A}'$, $|\tilde{A}'| \geq a_{\mathcal{F}}$. Note that, since $|\tilde{A}| \geq |\tilde{A}'|$, in this case we have that, for all such $\tilde{A}'$, $|\tilde{A}| \geq a_{\mathcal{F}}$. This justifies the arbitrary canonical partition enhancement in Step 3 and the fact that, if $|\tilde{A}| < a_{\mathcal{F}}$ in Step 3, then the algorithm reports a no-instance.

Let us now show the correctness of Step 2, and for this we do not suppose anymore that (G, k) is a yes-instance since the argument is the same for both types of instances. Suppose that the algorithm finds in Step 1 a set A' of size at most b and a regular flatness pair $(W_5, \mathfrak{R}_5)$ of $D_{W_4} \setminus A'$ of height r_5 whose $\mathfrak{R}_5$-compass has treewidth at most t. We obtain a 7-tuple $\mathfrak{R}_5'$ by adding all vertices of $G \setminus V(\mathsf{Compass}_{\mathfrak{R}_5}(W_5))$ to the set in the first coordinate of $\mathfrak{R}_5$. Since $(W_5, \mathfrak{R}_5)$ is a regular flatness pair of $D_{W_4} \setminus A'$ and since the vertices added in $\mathfrak{R}_5'$ are either in A, or adjacent at most to the perimeter of W_4, then $(W_5, \mathfrak{R}_5')$ is a regular flatness pair of $G \setminus (A \cup A')$. Since $\mathsf{Compass}_{\mathfrak{R}_5}(W_5) = \mathsf{Compass}_{\mathfrak{R}_5'}(W_5)$, $\mathsf{Compass}_{\mathfrak{R}_5'}(W_5)$ has treewidth at most t. Thus, if we apply the algorithm `Homogeneous` of Proposition 10 with input $(r_6, a+b, a+b, d, t, G, A \cup A', W_5, \mathfrak{R}_5')$, we obtain a flatness pair $(W_6, \mathfrak{R}_6)$ of $G \setminus (A \cup A')$ of height r_6 that is d-homogeneous with respect to $2^{A \cup A'}$ and is a W^*-tilt of $(W_5, \mathfrak{R}_5')$ for some subwall W^* of W_5. Since $|A \cup A'| \leq a + b$, for any set $X \subseteq V(G)$, $|A \setminus X| \leq a + b$. Since $G \setminus S \in \mathsf{exc}(\mathcal{F})$ and $\mathsf{bid}_{G\setminus(A\cup A'),W_6}(S) \leq |S| \leq k$, by applying the algorithm `Find-Irrelevant Vertex` of Proposition 11 with input $(k, a+b, G, A \cup A', W_6, \mathfrak{R}_6)$, we obtain a vertex v such that $G \setminus S \in \mathsf{exc}(\mathcal{F})$ if and only if $G \setminus (S \setminus v) \in \mathsf{exc}(\mathcal{F})$. It follows that (G, k) and $(G \setminus v, k)$ are indeed equivalent instances of $\mathcal{F}$-M-Deletion.

We now suppose that (G, k) is a no-instance. In the beginning of Step 1, the algorithm either reports a no-instance or finds a wall. In the latter case, the algorithm either goes to Step 2 or Step 3. If it runs Step 2, the previous paragraph justifies that the algorithm finds a vertex v such that $(G \setminus v, k)$ is a no-instance. If the algorithm runs Step 3, then it either reports a no-instance or recursively runs on instances $(G \setminus y, k-1)$. If $(G \setminus y, k-1)$ is yes-instance, then so is (G, k). Thus, $(G \setminus y, k-1)$ is a no-instance for every considered vertex y and the algorithm always reports a no-instance. Hence, Theorem 1 follows.

5 Concluding remarks

For a minor-closed graph class $\mathcal{G}$, we proved that Vertex Deletion to $\mathcal{G}$ can be solved in time $2^{\mathsf{poly}(k)} \cdot n^2$ and that Elimination Distance to $\mathcal{G}$ can be solved in time $2^{2^{2^{\mathsf{poly}(k)}}} \cdot n^2$, and in time $2^{2^{c \cdot k^2 \log k}} \cdot n^2$ and $2^{\mathsf{poly}(k)} \cdot n^3$ in the case where the obstruction set of $\mathcal{G}$ contains an apex-graph. Here the degree of poly and c heavily depend on the size of the obstructions of $\mathcal{G}$. An open question is whether $\mathsf{poly}(k)$ could be replaced by $c \cdot k^d$ for some constant c depending on $\mathcal{G}$ and some *universal* constant d (independent of $\mathcal{G}$). We tend to believe that this dependence on $\mathcal{G}$ in the exponent of the polynomial is unavoidable, at least if we want to use the irrelevant vertex technique, and specially our definition of homogeneity.

On the other hand, we are not aware, for any of the two considered problems, of any lower bound, assuming the Exponential Time Hypothesis [25], stronger than $2^{o(k)} \cdot n^{\mathcal{O}(1)}$, which follows quite easily from known results for Vertex Cover. Proving stronger lower bounds seems to be quite challenging.

Another open problem is whether it is possible to drop the time complexity of Elimination Distance to $\mathcal{G}$ to $2^{\mathsf{poly}(k)} \cdot n^2$ for *every* minor-closed graph class $\mathcal{G}$. We tend to believe that this should be possible. However, it seems to require to use branching ingeniously and, in particular, to find equivalent instances of Elimination Distance to $\mathcal{G}$ with a decreasing value of k.

As for the polynomial running time of our FPT-algorithms, a priori, nothing prevents the existence of algorithms running in *linear time*, although we are quite far from achieving this. Kawarabayashi [30] presented such a linear FPT-algorithm for the Planarization problem, heavily relying on the embedding on the resulting planar graph. Extending this technique to general minor-closed classes would require a very compact encoding of the (entangled) structure of minor-free graphs [48] that would be possible to handle in linear time.

References

1 Isolde Adler, Martin Grohe, and Stephan Kreutzer. Computing excluded minors. In *Proc. of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 641–650, 2008. URL: `http://dl.acm.org/citation.cfm?id=1347082.1347153`.

2 Akanksha Agrawal, Lawqueen Kanesh, Daniel Lokshtanov, Fahad Panolan, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. Deleting, Eliminating and Decomposing to Hereditary Classes Are All FPT-Equivalent. In *Proc. of the 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1976–2004, 2022. `doi:10.1137/1.9781611977073.79`.

3 Akanksha Agrawal, Lawqueen Kanesh, Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. An FPT algorithm for elimination distance to bounded degree graphs. In *Proc. of the 38th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 187 of *LIPIcs*, pages 5:1–5:11, 2021. `doi:10.4230/LIPIcs.STACS.2021.5`.

4 Akanksha Agrawal and M. S. Ramanujan. On the parameterized complexity of clique elimination distance. In *Proc. of the 15th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 180 of *LIPIcs*, pages 1:1–1:13, 2020. `doi:10.4230/LIPIcs.IPEC.2020.1`.

5 Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. A complexity dichotomy for hitting connected minors on bounded treewidth graphs: the chair and the banner draw the boundary. In *Proc. of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 951–970, 2020. `doi:10.1137/1.9781611975994.57`.

6 Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Hitting minors on bounded treewidth graphs. I. General upper bounds. *SIAM Journal on Discrete Mathematics*, 34(3):1623–1648, 2020. doi:10.1137/19M1287146.

7 Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Hitting minors on bounded treewidth graphs. II. Single-exponential algorithms. *Theoretical Computer Science*, 814:135–152, 2020. doi:10.1016/j.tcs.2020.01.026.

8 Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Hitting minors on bounded treewidth graphs. III. Lower bounds. *Journal of Computer and System Sciences*, 109:56–77, 2020. doi:10.1016/j.jcss.2019.11.002.

9 Hans L. Bodlaender, John R. Gilbert, Ton Kloks, and Hjálmtyr Hafsteinsson. Approximating treewidth, pathwidth, and minimum elimination tree height. In *Proc. of the 17th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 570 of *LNCS*, pages 1–12, 1991. doi:10.1007/3-540-55121-2_1.

10 Jannis Bulian and Anuj Dawar. Graph isomorphism parameterized by elimination distance to bounded degree. *Algorithmica*, 75(2):363–382, 2016. doi:10.1007/s00453-015-0045-3.

11 Jannis Bulian and Anuj Dawar. Fixed-parameter tractable distances to sparse graph classes. *Algorithmica*, 79(1):139–158, 2017. doi:10.1007/s00453-016-0235-7.

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

13 Reinhard Diestel. *Graph Theory*, volume 173. Springer-Verlag, 5th edition, 2017. doi:10.1007/978-3-662-53622-3.

14 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

15 Zdeněk Dvořák, Archontia C. Giannopoulou, and Dimitrios M. Thilikos. Forbidden graphs for tree-depth. *European Journal of Combinatorics*, 33(5):969–979, 2012. doi:10.1016/j.ejc.2011.09.014.

16 Eduard Eiben, Robert Ganian, Thekla Hamm, and O-joung Kwon. Measuring what matters: A hybrid approach to dynamic programming with treewidth. *Journal of Computer and System Sciences*, 121:57–75, 2021. doi:10.1016/j.jcss.2021.04.005.

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

18 Fedor V. Fomin, Petr A. Golovach, and Dimitrios M. Thilikos. Parameterized complexity of elimination distance to first-order logic properties. *ACM Transactions on Computational Logic*, 23(3):17:1–17:35, 2022. doi:10.1145/3517129.

19 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar $\mathcal{F}$-deletion: Approximation, kernelization and optimal FPT algorithms. In *Proc. of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 470–479, 2012. doi:10.1109/FOCS.2012.62.

20 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Hitting topological minors is FPT. In *Proc. of the 52nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 1317–1326, 2020. doi:10.1145/3357713.3384318.

21 Archontia C. Giannopoulou, O-joung Kwon, Jean-Florent Raymond, and Dimitrios M. Thilikos. Lean tree-cut decompositions: Obstructions and algorithms. In *Proc. of the 36th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 126 of *LIPIcs*, pages 32:1–32:14, 2019. doi:10.4230/LIPIcs.STACS.2019.32.

22 Archontia C. Giannopoulou, Michal Pilipczuk, Jean-Florent Raymond, Dimitrios M. Thilikos, and Marcin Wrochna. Linear kernels for edge deletion problems to immersion-closed graph classes. In *Proc. of the 44th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 80 of *LIPIcs*, pages 57:1–57:15, 2017. doi:10.4230/LIPIcs.ICALP.2017.57.

23 Archontia C. Giannopoulou, Michal Pilipczuk, Jean-Florent Raymond, Dimitrios M. Thilikos, and Marcin Wrochna. Cutwidth: Obstructions and algorithmic aspects. *Algorithmica*, 81(2):557–588, 2019. doi:10.1007/s00453-018-0424-7.

24 Jiong Guo, Falk Hüffner, and Rolf Niedermeier. A structural view on parameterizing problems: Distance from triviality. In *Proc. of the 1st International Workshop on Parameterized and Exact Computation (IWPEC)*, volume 3162 of *LNCS*, pages 162–173, 2004. doi:10.1007/978-3-540-28639-4_15.

25 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

26 Bart M. P. Jansen, Jari J. H. de Kroon, and Michał Włodarczyk. Vertex deletion parameterized by elimination distance and even less. In *Proc. of the 53rd Annual ACM-SIGACT Symposium on Theory of Computing (STOC)*, pages 1757–1769, 2021. doi:10.1145/3406325.3451068.

27 Bart M. P. Jansen, Daniel Lokshtanov, and Saket Saurabh. A near-optimal planarization algorithm. In *Proc. of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1802–1811, 2014. doi:10.1137/1.9781611973402.130.

28 Mamadou Moustapha Kanté and O-joung Kwon. An upper bound on the size of obstructions for bounded linear rank-width, 2014. arXiv:1412.6201.

29 Mamadou Moustapha Kanté and O-joung Kwon. Linear rank-width of distance-hereditary graphs II. vertex-minor obstructions. *European Journal of Combinatorics*, 74:110–139, 2018. doi:10.1016/j.ejc.2018.07.009.

30 Ken-ichi Kawarabayashi. Planarity allowing few error vertices in linear time. In *Proc. of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 639–648, 2009. doi:10.1109/FOCS.2009.45.

31 Ken-ichi Kawarabayashi, Yusuke Kobayashi, and Bruce A. Reed. The disjoint paths problem in quadratic time. *Journal of Combinatorial Theory, Series B*, 102(2):424–435, 2012. doi:10.1016/j.jctb.2011.07.004.

32 Ken-ichi Kawarabayashi, Robin Thomas, and Paul Wollan. A new proof of the flat wall theorem. *Journal of Combinatorial Theory, Series B*, 129:204–238, 2018. doi:10.1016/j.jctb.2017.09.006.

33 Ken-ichi Kawarabayashi and Paul Wollan. A Shorter Proof of the Graph Minor Algorithm: The Unique Linkage Theorem. In *Proc. of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 687–694, 2010. doi:10.1145/1806689.1806784.

34 Eun Jung Kim, Alexander Langer, Christophe Paul, Felix Reidl, Peter Rossmanith, Ignasi Sau, and Somnath Sikdar. Linear kernels and single-exponential algorithms via protrusion decompositions. *ACM Transactions on Algorithms*, 12(2):21:1–21:41, 2016. doi:10.1145/2797140.

35 Eun Jung Kim, Maria J. Serna, and Dimitrios M. Thilikos. Data-compression for parametrized counting problems on sparse graphs. In *Proc. of the 29th International Symposium on Algorithms and Computation (ISAAC)*, volume 123 of *LIPIcs*, pages 20:1–20:13, 2018. doi:10.4230/LIPIcs.ISAAC.2018.20.

36 Ton Kloks. *Treewidth, Computations and Approximations*, volume 842 of *Lecture Notes in Computer Science*. Springer, 1994. doi:10.1007/BFb0045375.

37 Tomasz Kociumaka and Marcin Pilipczuk. Deleting Vertices to Graphs of Bounded Genus. *Algorithmica*, 81(9):3655–3691, 2019. doi:10.1007/s00453-019-00592-7.

38 Jens Lagergren. An upper bound on the size of an obstruction. In *Graph Structure Theory*, volume 147 of *Contemporary Mathematics*, pages 601–621. American Mathematical Society, 1991. doi:10.1090/conm/147/01202.

39 Jens Lagergren. Upper bounds on the size of obstructions and intertwines. *Journal of Combinatorial Theory, Series B*, 73:7–40, 1998. doi:10.1006/jctb.1997.1788.

40 Jens Lagergren and Stefan Arnborg. Finding minimal forbidden minors using a finite congruence. In *Proc. of the 18th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 510 of *LNCS*, pages 532–543, 1991. doi:10.1007/3-540-54233-7_161.

41 John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980. doi:10.1016/0022-0000(80)90060-4.

42 Dániel Marx and Ildikó Schlotter. Obtaining a planar graph by vertex deletion. *Algorithmica*, 62(3-4):807–822, 2012. doi:10.1007/s00453-010-9484-z.

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

44 Rolf Niedermeier. *Invitation to fixed parameter algorithms*, volume 31. Oxford University Press, 2006. doi:10.1093/ACPROF:OSO/9780198566076.001.0001.

45 Alex Pothen. The complexity of optimal elimination trees. *Technical Report. Pennsylvania State University. Dept. of Computer Science*, 1988. URL: https://www.cs.purdue.edu/homes/apothen/Papers/shortest-etree1988.pdf.

46 Felix Reidl, Peter Rossmanith, Fernando Sánchez Villaamil, and Somnath Sikdar. A faster parameterized algorithm for treedepth. In *Proc. of the 41st International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 8572 of *LNCS*, pages 931–942, 2014. doi:10.1007/978-3-662-43948-7_77.

47 Neil Robertson and Paul D. Seymour. Graph Minors. XIII. The Disjoint Paths Problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995. doi:10.1006/jctb.1995.1006.

48 Neil Robertson and Paul D. Seymour. Graph Minors. XVI. Excluding a non-planar graph. *Journal of Combinatorial Theory, Series B*, 89(1):43–76, 2003. doi:10.1016/S0095-8956(03)00042-X.

49 Neil Robertson and Paul D. Seymour. Graph Minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004. doi:10.1016/j.jctb.2004.08.001.

50 Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. A more accurate view of the Flat Wall Theorem, 2021. arXiv:2102.06463.

51 Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. k-apices of minor-closed graph classes. II. Parameterized algorithms. *ACM Transactions on Algorithms*, 18(3):21:1–21:30, 2022. Short version in *Proc. of the 47th International Colloquium on Automata, Languages and Programming (ICALP), volume 168 of LIPIcs, pages 95:1-95:20, 2020.* doi:10.1145/3519028.

52 Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. k-apices of minor-closed graph classes. I. Bounding the obstructions. *Journal of Combinatorial Theory, Series B*, 161:180–227, 2023. doi:10.1016/j.jctb.2023.02.012.

53 Andrew Thomason. The extremal function for complete minors. *Journal of Combinatorial Theory, Series B*, 81(2):318–338, 2001. doi:10.1006/jctb.2000.2013.