Kernelization for feedback vertex set via elimination distance to a forest

Discrete Applied Mathematics 346 (2024) 192–214

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Kernelization for feedback vertex set via elimination distance to a forest[☆]

David J.C. Dekker, Bart M.P. Jansen [*]

Eindhoven University of Technology, PO Box 513, Eindhoven, 5600 MB, Noord Brabant, The Netherlands

ARTICLE INFO

Article history:
Received 29 March 2023
Received in revised form 13 September 2023
Accepted 14 December 2023
Available online 23 December 2023

MSC:
68Q25
68Q27
05C85

Keywords:
Feedback vertex set
Kernelization
Elimination distance

ABSTRACT

We study efficient preprocessing for the undirected Feedback Vertex Set problem, a fundamental problem in graph theory which asks for a minimum-sized vertex set whose removal yields an acyclic graph. More precisely, we aim to determine for which parameterizations this problem admits a polynomial kernel. While a characterization is known for the related Vertex Cover problem based on the recently introduced notion of bridge-depth, it remained an open problem whether this could be generalized to Feedback Vertex Set. The answer turns out to be negative; the existence of polynomial kernels for structural parameterizations for Feedback Vertex Set is governed by the elimination distance to a forest. Under the standard assumption NP $\not\subseteq$ coNP/poly, we prove that for any minor-closed graph class $\mathcal{G}$, Feedback Vertex Set parameterized by the size of a modulator to $\mathcal{G}$ has a polynomial kernel if and only if $\mathcal{G}$ has bounded elimination distance to a forest. This captures and generalizes all existing kernels for structural parameterizations of the Feedback Vertex Set problem.

1. Introduction

For NP-complete problems, a polynomial-time algorithm solving any problem instance exactly is unlikely to exist. However, as one is often interested in solving specific instances, one can try to exploit characteristics of problem instances and develop algorithms that are fast when the input has certain properties. We therefore associate a parameter with each problem instance. In our context, a problem instance is a graph for which we ask for the existence of a vertex set of size at most ℓ having certain properties. Such a parameterized instance can be denoted with a triple (G, ℓ, k), where we are asking for the existence of a solution of size at most ℓ for a graph G with parameter k. We say that an algorithm is

☆ This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 803421, ReduceSearch).

An extended abstract of this work appeared under the same title in the proceedings of the 48th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2022.

* Corresponding author.
E-mail addresses: d.j.c.dekker@student.tue.nl (D.J.C. Dekker), b.m.p.jansen@tue.nl (B.M.P. Jansen).

https://doi.org/10.1016/j.dam.2023.12.016

fixed-parameter tractable (FPT) for such a parameterization if it solves any instance (G, ℓ, k) of size n, as described above, in time bounded by $f(k)n^{\mathcal{O}(1)}$ for some computable function $f : \mathbb{N} \to \mathbb{N}$.

A strongly related field is that of *kernelization*. This field focuses on reducing a parameterized instance (G, ℓ, k) in polynomial time to an equivalent instance (G', ℓ', k') whose size is bounded by a computable function of the parameter. We speak of a polynomial kernel when this function is a polynomial. It is known that a decidable parameterized problem is fixed-parameter tractable if and only if it admits a kernelization [4]. In our quest for determining which parameterizations enable efficient algorithms, it is therefore interesting to determine those that allow a polynomial kernel.

This paper focuses on polynomial kernels for the undirected Feedback Vertex Set problem, which is an NP-complete problem in graph theory as originally identified by Karp [34]. For an undirected graph G, a vertex set $X \subseteq V(G)$ is a *feedback vertex set* if the graph is acyclic after removal of X. We call a vertex set whose removal yields a graph in some graph class $\mathcal{G}$ a *$\mathcal{G}$-modulator* and define the deletion distance to $\mathcal{G}$ as its minimum size. The Feedback Vertex Set problem now asks for the minimum size of a feedback vertex set, or equivalently, the deletion distance to a forest. For a graph G, we let $\textsc{fvs}(G)$ (the *feedback vertex number* of G) denote that minimum size. Our main question is for which parameterizations the Feedback Vertex Set problem admits a polynomial kernel.

Before exploring the Feedback Vertex Set problem further, we should mention the related Vertex Cover problem. It asks for a minimum set of vertices hitting all edges in a graph. While a kernel in the solution size with a linear number of vertices can be obtained using various techniques [1,13–15,40], a polynomial kernel in a structurally smaller parameter was only discovered in 2011, when Jansen and Bodlaender developed a polynomial kernel in the feedback vertex number of a graph [30]. From there, many polynomial kernels for Vertex Cover were discovered in modulators to even larger graph classes [9,22,26,39]. Bougeret, Jansen and Sau recently proved the following characterization under common hardness assumptions: Vertex Cover admits a polynomial kernel in the size of a modulator to a minor-closed graph family $\mathcal{G}$ if and only if $\mathcal{G}$ has bounded *bridge-depth* [8]. With this result, they generalized all existing work on kernels in the size of modulators to minor-closed graph families, and they proved that their results cannot be improved further under common hardness assumptions.

For Feedback Vertex Set, the first polynomial kernel (parameterized by solution size k) with size bound $\mathcal{O}(k^{11})$ was obtained in 2006 and it was subsequently improved to a quadratic kernel [7,12,42]. After the improvements for Vertex Cover, researchers also tried to develop polynomial kernels in smaller parameters for Feedback Vertex Set [32,33,38]. It remained an open problem whether these results could be generalized further or whether there exists some parameter that characterizes Feedback Vertex Set similarly to how bridge-depth characterizes Vertex Cover. In particular, Bougeret, Jansen and Sau suggested in their paper on Vertex Cover that the deletion distance to constant bridge-depth might also be an interesting parameter to consider for problems such as Feedback Vertex Set. We therefore aim to answer the question for which graph families $\mathcal{G}$ the Feedback Vertex Set problem admits a polynomial kernel when parameterized by the size of a $\mathcal{G}$-modulator.

Our results. To our initial surprise, the results for Vertex Cover cannot be generalized to Feedback Vertex Set. It turns out that a minor-closed graph family $\mathcal{G}$ must have bounded *elimination distance to a forest* (Definition 1), in order to allow a polynomial kernel in a $\mathcal{G}$-modulator. The elimination distance to a forest is the minimum number of rounds needed to transform the graph into a forest when removing one vertex from each connected component in each round. The concept of elimination distance to a graph class $\mathcal{G}$ was introduced by Bulian and Dawar [11] and is another generalization of the more common parameter treedepth [41]. Our main result is the following.

Theorem 1. *Let $\mathcal{G}$ be a minor-closed graph family and assume* NP $\not\subseteq$ coNP/poly. *Then* Feedback Vertex Set *admits a polynomial kernel in the size of a $\mathcal{G}$-modulator if and only if $\mathcal{G}$ has bounded elimination distance to a forest.*

The minor-closed and hardness assumptions are only needed for the lower bound. To the best of our knowledge, our kernel generalizes all known polynomial kernels for the Feedback Vertex Set problem. Both the kernel and its correctness proof follow the structure of the kernel for $\mathcal{F}$-Minor Free Deletion in the deletion distance to a graph of constant treedepth by Jansen and Pieterse [32]. The correctness proof of their kernel crucially relies on their Lemma 3 whose technical proof spans thirty pages. We require a variation of this lemma. On the one hand, our variation is more involved since it deals with elimination distance to a forest rather than treedepth; on the other hand, it is simpler since it concerns only Feedback Vertex Set rather than $\mathcal{F}$-Minor Free Deletion. As a result of this simplification, we can formulate the lemma without the use of minors. Roughly speaking, the lemma says that in a graph G of bounded elimination distance to a forest, if no minimum feedback vertex set exists which simultaneously hits a prescribed set of *partial* cycles (single vertices in a set S or paths between two terminals in a set T), then the same holds for some sets $S^* \subseteq S$ and $T^* \subseteq T$ of constant size. As shown in previous work, this limited sensitivity with respect to whether optimal solutions can break all partial forbidden structures is crucial for the kernelization complexity. As one of our main contributions, we prove this lemma using a strategy that differs significantly from the one followed in earlier work [32].

Lemma 1. *Let G be a connected graph with disjoint vertex sets $S, T \subseteq V(G)$. Suppose that any minimum feedback vertex set X of G misses some vertex from S or leaves two vertices from T connected in $G - X$. Then there exist sets $S^* \subseteq S$ and $T^* \subseteq T$ whose sizes only depend on the elimination distance to a forest of G, such that any minimum feedback vertex set X of G misses some vertex from S^* or leaves two vertices from T^* connected in $G - X$.*

Once Lemma 1 is proven, the kernelization upper bounds follow similarly to earlier work [32]. As for the lower bound in Theorem 1, we are also able to generalize our proof for other $\mathcal{F}$-Minor Free Deletion problems as described in Theorem 2.

Theorem 2. *Let $\mathcal{G}$ be a minor-closed family of graphs and let $\mathcal{F}$ be a finite set of biconnected planar graphs on at least three vertices. If $\mathcal{G}$ has unbounded elimination distance to an $\mathcal{F}$-minor free graph, then $\mathcal{F}$-Minor Free Deletion does not admit a polynomial kernel in the size of a $\mathcal{G}$-modulator, unless* NP $\subseteq$ coNP/poly.

Related work. FPT algorithms for Feedback Vertex Set parameterized by the solution size are known with running times $\mathcal{O}^*(3.490^k)$ (deterministic) and $\mathcal{O}^*(2.7^k)$ (randomized) [28,35,37]. The parameterization by treewidth has led to deterministic $\mathcal{O}^*(11.36^k)$ and randomized $\mathcal{O}^*(3^k)$ algorithms [5,18]. Other FPT algorithms include parameterizations by deletion distance to a chordal graph, by cliquewidth and by the number of vertices of degree at least 4 [10,29,33,38]. Recently, Donkers and Jansen proposed a parameterization based on the complexity of the solution structure [20]. They introduced the notion of *antlers*, similar to the notion of crown decompositions for Vertex Cover.

Organization. Section 2 introduces all relevant terminology. Section 3 presents our kernel and thereby proves the 'if' direction of Theorem 1. Then Section 4 contains the proof of Theorem 2, thereby also proving the 'only if' direction of Theorem 1. Lastly, Section 5 contains our conclusions and discusses future work.

2. Preliminaries

For a positive integer n, we use the shorthand $[n]$ for the set of all natural numbers i with $1 \leq i \leq n$. All graphs we consider are finite, undirected and simple. When G is a graph, we let $V(G)$ denote the vertex set of G and $E(G)$ the edge set. For $S \subseteq V(G)$, the graph $G - S$ is the graph where all vertices in S and all incident edges are removed, and the graph $G[S]$ is the subgraph of G induced by the vertices in S. When an edge exists between two vertices in G, we say that these vertices are *adjacent*. The *neighbors* of v in G, denoted with $N_G(v)$, are the vertices adjacent to a vertex $v \in V(G)$ in G. For $S \subseteq V(G)$, we say that $v \in V(G - S)$ is adjacent to S if there exists some edge between v and a vertex in S. The set $N_G(S)$ contains all vertices $v \in V(G - S)$ for which this holds. We will sometimes slightly abuse notation and speak of a vertex being adjacent to some subgraph, rather than to the vertices in that subgraph. We say that two vertices are *connected* in G when they are in the same connected component. The set $\text{cc}(G)$ denotes the set of connected components (or shortly components) of G; hence $\text{cc}(G)$ is a set of graphs. For sets $S, T \subseteq (V)$, we say that S *separates* T if each component of $G - S$ contains at most one vertex from T. Notice that we do not require S and T to be disjoint. Such a set S is a *vertex multiway cut* of T in G. We use the notation $\mathcal{O}_\eta(f(n))$ to describe the functions in η and n which can be bounded by $g(\eta) \cdot f(n)$ for some computable function g.

A concept that will be used extensively is that of graph minors, which uses the notion of *edge contraction*. When uv is an edge in a graph G, contracting this edge replaces vertices u and v by a new vertex whose set of neighbors is $N_G(\{u, v\})$. Now H is a *minor* of G if H can be obtained from G by removing vertices, removing edges and contracting edges. Alternatively, one can define H to be a minor of G if there exists a *minor model* $\phi : V(H) \to 2^{V(G)}$, such that for any $v \in V(H)$ the graph $G[\phi(u)]$ is connected, for any distinct $u, v \in V(H)$ we have $\phi(u) \cap \phi(v) = \emptyset$, and for any edge uv in H there exists an edge between a vertex in $\phi(u)$ and a vertex in $\phi(v)$ in G.

A graph G has an *$\mathcal{H}$-minor* for a set of graphs $\mathcal{H}$ if G contains some graph $H \in \mathcal{H}$ as a minor. For a minor model ϕ of H in G and a set $S \subseteq V(H)$, we use the shorthand notation $\phi(S) := \bigcup_{v \in S} \phi(v)$. We say that a minor model ϕ of H in G is *minimal*, if there does not exist a minor model ϕ' of H in G with $\phi'(V(H)) \subsetneq \phi(V(H))$. A minor model ϕ of H in G and a minor model ϕ' of H' in G *intersect* if $\phi(V(H)) \cap \phi'(V(H')) \neq \emptyset$. A graph H is a *proper minor* of a graph G if H is a minor of G but H is not isomorphic to G.

2.1. Elimination distance and treewidth

Our work relies crucially on the concept of elimination distance as introduced by Bulian and Dawar [11].

Definition 1 (*Elimination Distance*). Let G be a graph and $\mathcal{G}$ a graph family. Then the elimination distance of G to $\mathcal{G}$ is

$$\text{ED}_{\mathcal{G}}(G) = \begin{cases} 0 & \text{if } G \in \mathcal{G}, \\ \max_{G' \in \text{cc}(G)} \text{ED}_{\mathcal{G}}(G') & \text{if } |\text{cc}(G)| > 1, \\ \min_{v \in V(G)} \text{ED}_{\mathcal{G}}(G - \{v\}) + 1 & \text{otherwise.} \end{cases}$$

We only consider graph families $\mathcal{G}$ that are minor-closed. We use the shorthand $\mathcal{G}_F$ for the graph family containing precisely all forests. We say that a graph family $\mathcal{G}$ has bounded elimination distance to some graph class $\mathcal{H}$, if there is a universal constant c such that $\text{ED}_{\mathcal{H}}(G) \leq c$ for all $G \in \mathcal{G}$. The elimination distance of a graph G to the empty graph is called the *treedepth* of G and is denoted with $\text{TD}(G)$. More intuitively, the elimination distance to a graph class $\mathcal{G}$ can be interpreted as the minimum number of 'elimination iterations' that are necessary to obtain a graph where every connected component is in $\mathcal{G}$. In such an iteration, one is allowed to remove one vertex from each connected component. This interpretation leads to the notion of an elimination forest.

Definition 2 ($\mathcal{G}$*-elimination forest*)**.** Let G be a graph and $\mathcal{G}$ a graph family. A $\mathcal{G}$*-elimination forest* of G is a tuple $\left(F, (B_u)_{u \in V(F)}\right)$ where F is a rooted forest and where each vertex $v \in V(F)$ has a bag $B_v \subseteq V(G)$ such that:

- The bags define a partition of $V(G)$, i.e. for any vertex $v \in V(G)$ there is a unique node $u \in V(F)$ with $v \in B_u$.
- For any non-leaf u of F, its bag B_u contains precisely one vertex.
- For any leaf u of F, the graph $G[B_u]$ is connected and $G[B_u] \in \mathcal{G}$.
- For any edge uv in G, let $s, t \in V(F)$ be the nodes such that $u \in B_s$ and $v \in B_t$. Then s is an ancestor of t, or t is an ancestor of s in F.

We define the height of an elimination forest F to be the maximum number of edges on a path from the root to a leaf in a component of F. A simple induction shows that $\text{ED}_{\mathcal{G}}(G)$ is equal to the minimum height of a $\mathcal{G}$-elimination forest of G.

We will use these elimination forests extensively for our kernel and therefore introduce some shorthand notation. Let $(F, (B_u)_{u \in V(F)})$ be a $\mathcal{G}$-elimination forest. Let v be a vertex in F. The *tail* of v, denoted with $\text{TAIL}(v)$, is defined as the union of B_u over all proper ancestors u of v. The closed tail $\text{TAIL}[v]$ also includes B_v. Similarly, $\text{TREE}(v)$ denotes the union of B_u over all proper descendants u of v and $\text{TREE}[v]$ also includes B_v. The subgraph of G induced by all vertices in $\text{TREE}[v]$ is denoted with G_v. We will sometimes slightly abuse notation and use G_v as a vertex set. We use the shorthand G_v^+ to describe the induced subgraph on the vertices in $\text{TREE}[v] \cup \text{TAIL}[v]$. Note that, as a consequence of Definition 2, for any leaf v of F we have $N(B_v) \subseteq \text{TAIL}(v)$.

We will also introduce the notion of *bridge-depth* as introduced by Bougeret, Jansen and Sau [8]. A *bridge* in a graph G is an edge whose removal increases the number of connected components of G. The concept of bridge-depth now allows us to delete a set of vertices S as long as $G[S]$ is connected and each edge in $G[S]$ is a bridge in G. Such a structure $G[S]$ is called a *tree of bridges*. Observe that a single vertex is always a tree of bridges.

Definition 3 (*Bridge-Depth*)**.** Let G be a graph. The bridge-depth of G is defined as

$$\text{BD}(G) = \begin{cases} 0 & \text{if } G \text{ is the empty graph,} \\ \max\limits_{G' \in \text{cc}(G)} \text{BD}(G') & \text{if } |\text{cc}(G)| > 1, \\ \min\limits_{\substack{S \subseteq V(G): \\ G[S] \text{ is a tree of bridges}}} \text{BD}(G - S) + 1 & \text{otherwise.} \end{cases}$$

Cf. [8] for equivalent definitions. Lastly, we sometimes use the more common concept of *treewidth* which can be defined using a *tree decomposition*.

Definition 4 (*Tree Decomposition*)**.** Let G be a graph. A tree decomposition of G is a tuple $(R, (B_u)_{u \in V(R)})$ where R is a rooted tree and where each node $u \in V(R)$ has a bag $B_u \subseteq V(G)$ with the following properties:

- For every vertex $v \in V(G)$, there exists a node $u \in V(R)$ with $v \in B_u$.
- For every edge $vw \in E(G)$, there exists a node $u \in V(R)$ with $\{v, w\} \subseteq B_u$.
- For every vertex $u \in V(G)$, the subgraph of R induced by all vertices whose bag contains u is connected.

For $u \in V(R)$ in the setting above, we let R_u denote the subtree of R rooted at u. Let G_u be the subgraph of G induced by the union of B_v over all descendants v of u in the rooted tree R, i.e.

$$G_u = G\left[\bigcup_{v \in V(R_u)} B_v\right].$$

The *width* of a tree decomposition is defined as the size of its largest bag minus one. The *treewidth* of a graph G, denoted with $\text{TW}(G)$, can now be defined as the minimum width over all tree decompositions of G. We mention some useful properties of these concepts in Proposition 1.

Proposition 1. *Let G be a graph with $\mathcal{G}_F$-elimination forest $\left(F, (B_u)_{u \in V(F)}\right)$ of height η. Let Y be a minimum feedback vertex set in G and let v be a leaf in F. Then the following claims hold.*

1. $\text{TW}(G) \leq \text{BD}(G) \leq \text{ED}_{\mathcal{G}_F}(G) + 1$.
2. *Y contains at most η vertices from B_v.*
3. *Any path in G from a vertex in B_v to a vertex outside B_v contains a vertex in $\text{TAIL}(v)$.*
4. *If $u \in V(F)$, then u has at most η children w for which $Y \cap G_w$ is not a minimum feedback vertex set in G_w.*

Proof. Proof of 1. Observe that a single vertex is also a tree of bridges. One can therefore derive that $\text{BD}(G) \leq \text{ED}_{\mathcal{G}_F}(G)+1$. There is a difference of one, since a forest has elimination distance 0 to a forest while its bridge-depth is one. Besides, it is known that $\text{TW}(G) \leq \text{BD}(G)$ for any graph G [8, Proposition 3.2], thereby concluding the proof.

Proof of 2. Suppose towards a contradiction that Y is a minimum feedback vertex set in G, but $|B_v \cap Y| > \eta$. We claim that $Y' := (Y \setminus B_v) \cup \text{TAIL}(v)$ is a feedback vertex set in G of smaller size. Since $|\text{TAIL}(v)| \leq \eta$, it follows that $|Y'| < |Y|$.

Furthermore, suppose towards a contradiction that Y' is not a feedback vertex set, then $G-Y'$ contains some cycle disjoint from $\textsc{tail}(v)$. However, then this cycle must also be disjoint from B_v, as $G[B_v]$ is a forest and $N(B_v) \subseteq \textsc{tail}(v)$ which are all in Y'. Therefore, this cycle must also be contained in $G - Y$ which is a contradiction with Y being a feedback vertex set. We conclude that any minimum feedback vertex set contains at most η vertices from B_v.

Proof of 3. If there exists path from a vertex in B_v to a vertex outside B_v, then there also exists an edge between a vertex in B_v to a vertex outside B_v. By definition of an elimination forest and since v is a leaf, this edge has an endpoint in the bag of a proper ancestor of v, so it is in $\textsc{tail}(v)$.

Proof of 4. Let C denote the set of children of u and let $C' \subseteq C$ be those children w where $Y \cap G_w$ is not a minimum feedback vertex set in G_w. If $C' \neq \emptyset$, then u is not a leaf and therefore $|\textsc{tail}[u]| \leq \eta$. Assume towards a contradiction that $|C'| > \eta$. We will construct a feedback vertex set strictly smaller than Y. For all children $w \in C'$, let Y'_w be a minimum feedback vertex set in G_w. We now claim that

$$Y' := \left(Y \setminus \bigcup_{w \in C'} (Y \cap G_w)\right) \cup \textsc{tail}[u] \cup \bigcup_{w \in C'} Y'_w$$

is a feedback vertex set in G with less vertices than Y. Since $|Y \cap G_w| > |Y'_w|$ for each $w \in C'$, $|C'| > \eta$ and $|\textsc{tail}[u]| \leq \eta$, it follows that $|Y'| < |Y| - \eta + \eta = |Y|$.

Observe that any cycle that does not contain vertices in G_w for some child $w \in C'$ must be broken by Y', as no vertex from this cycle was removed from Y while Y is a feedback vertex set. Similarly, any cycle that only contains vertices in G_w for some $w \in C'$ must be broken by Y, because Y'_w is a feedback vertex set in G_w. Any remaining cycle in $G - Y'$ therefore must contain a vertex in G_w for some $w \in C'$ and a vertex outside this subgraph. By (3), any path in this component between a vertex in G_w and a vertex outside G_w must contain a vertex from $\textsc{tail}(w) = \textsc{tail}[u]$. This implies directly that also any cycle of the last category must be broken by Y'. We can conclude that Y' is a feedback vertex set with strictly fewer vertices than Y, thereby contradicting that Y is a minimum feedback vertex set. □

3. Kernelization upper bounds

We present a kernelization algorithm for Feedback Vertex Set parameterized by the deletion distance to a graph of constant elimination distance to a forest. The structure follows the kernelization algorithm for the generalized $\mathcal{F}$-Minor Free Deletion problem parameterized by deletion distance to a graph of constant treedepth, as presented by Jansen and Pieterse [32, Lemma 3]. The proof of their algorithm requires a lemma on labeled minors. Since we restrict ourselves to the Feedback Vertex Set problem, our reformulation in Lemma 1 in the introduction suffices for our purposes. We repeat and prove this formulation in Section 3.2 and describe the kernelization algorithm in Section 3.3. However, we start by deriving some properties of optimal feedback vertex sets in Section 3.1.

3.1. The structure of solutions with respect to a modulator

Let G be a graph and $X \subseteq V(G)$. For our kernel, it will be important to bound the number of relevant components of $G-X$ when X is a modulator to a graph with constant elimination distance to a forest. This will be done by characterizing how a component $C \in \text{cc}(G - X)$ is connected to X. To this end, we define the following notation.

Definition 5 (*$\mathcal{S}_G$ and $\mathcal{T}_G$*)**.** Let G be a graph with $X \subseteq V(G)$ and $C \in \text{cc}(G - X)$. Then

$$\mathcal{S}_G(C, X) := \{v \in V(C) \mid |N_G(v) \cap X| \geq 2\},$$
$$\mathcal{T}_G(C, X) := \{v \in V(C) \mid |N_G(v) \cap X| = 1\}.$$

These sets characterize how C is connected to the remainder of G. The vertices in $\mathcal{S}_G(C, X)$ are those vertices in C with at least two neighbors in X, while the vertices in $\mathcal{T}_G(C, X)$ are those with exactly one neighbor. We will often use these sets in the context of a feedback vertex set missing vertices from $\mathcal{S}_G(C, X)$ or leaving a pair in $\mathcal{T}_G(C, X)$ connected. We therefore use the notation with $\mathcal{S}$ and $\mathcal{T}$ in analogy with sets S and T in the formulation of Lemma 1. The following proposition presents such a relation between these sets and feedback vertex sets. Recall that we say that a vertex set X separates a vertex set T in a graph G if each connected component of $G - X$ contains at most one vertex from T.

Proposition 2. *Let G be a graph, $X \subseteq V(G)$ and $C^* \in \text{cc}(G - X)$. Define $\widehat{G} = G - V(C^*)$. Let $\widehat{Y}$ be a feedback vertex set in $\widehat{G}$ and let Y^* be a feedback vertex set in C^*. Define $X_0 = X \setminus \widehat{Y}$. If Y^* contains $\mathcal{S}_G(C^*, X_0)$ and separates $\mathcal{T}_G(C^*, X_0)$, then $\widehat{Y} \cup Y^*$ is a feedback vertex set in G.*

One should observe that this statement is slightly stronger than requiring that Y^* contains $\mathcal{S}_G(C^*, X)$ and separates $\mathcal{T}_G(C^*, X)$. We will rely on our weakened assumption when applying Proposition 2. Intuitively, the lemma states that the feedback vertex set Y^* in the component C^* combines with the feedback vertex set $\widehat{Y}$ of the remaining graph into a valid feedback vertex set for G, as long as Y^* is guaranteed to break potential cycles that intersect both C^* and $G - V(C^*)$.

Proof of Proposition 2. Assume towards a contradiction that $\widehat{Y} \cup Y^*$ is not a feedback vertex set in G. Then observe that any cycle must contain vertices both from C^* and from $\widehat{G}$, since any other cycle is broken trivially by either $\widehat{Y}$ or Y^*. This implies that there exists a path on at least two vertices in $C^* - Y^*$ whose endpoints have a neighbor in X_0: the path cannot contain only one vertex, since such a vertex would have two neighbors in X, while $\mathcal{S}_G(C^*, X_0)$ is included in Y^*. Therefore, the path has at least two vertices and both endpoints are in $\mathcal{T}_G(C^*, X_0)$. However, then a pair of vertices in $\mathcal{T}_G(C^*, X_0)$ is connected in $C^* - Y^*$, reaching a contradiction with the assumption that Y^* separates $\mathcal{T}_G(C^*, X_0)$. It follows that $G - (Y^* \cup \widehat{Y})$ is acyclic and that $Y^* \cup \widehat{Y}$ is a feedback vertex set in G. □

The following proposition explains a property of forests which is useful in the context of feedback vertex sets.

Proposition 3. *Let G be a forest with $X \subseteq V(G)$ and let $\mathcal{C} = \mathrm{cc}(G - X)$. Let $X^* \subseteq X$. There are at most $|X^*| - 1$ components $C \in \mathcal{C}$ for which there exist distinct $u, v \in X^*$ such that some u–v path contains a vertex in C.*

Proof. Assume towards a contradiction that there exist $|X^*|$ of these components. For such a component C, let P_C be a minimal path that has its endpoints in X^* while it also contains a vertex in C. Observe that such a path contains at least three vertices and notice that the minimality of the path implies that all vertices other than the endpoints are in C. If two of these paths have the same endpoints, we directly obtain a cycle in G consisting of these two paths. Otherwise, we create an auxiliary graph H on vertex set X^*. We add an edge between $z_1, z_2 \in X^*$ if for some component C the path P_C has endpoints z_1 and z_2. Since we add $|X^*|$ unique edges to a graph of at most $|X^*|$ vertices, H contains a cycle. Now consider the cyclic walk in G traversing all paths corresponding to the edges in the cycle in H. We claim that each vertex occurs only once in this walk. Observe that no vertex in X^* occurs twice, as only the endpoints of each path corresponding to a child are in Z. Furthermore, a vertex outside X^* cannot occur on multiple paths since those paths contain vertices from different components. Therefore all vertices are unique in the walk, so we obtain a cycle in G which contradicts it being acyclic. We conclude that there exist at most $|X^*| - 1$ of these components. □

The following proposition will be used to argue that for a modulator $X \subseteq V(G)$ in a graph G, there are only few components C of $G - X$ whose potential for making cycles with the rest of the graph is not completely broken by any given feedback vertex set Y. When formulating a reduction rule, we will only have to worry about the components that can potentially form a cycle together with other components of $G - X$.

Proposition 4. *Let G be a graph, $X \subseteq V(G)$ and $\mathcal{C} = \mathrm{cc}(G - X)$. Let Y be a feedback vertex set in G and let $X^* \subseteq X \setminus Y$. Then there are at most $|X^*| - 1$ components $C \in \mathcal{C}$ for which $Y \cap V(C)$ misses a vertex in $\mathcal{S}_G(C, X^*)$ or leaves a pair of vertices in $\mathcal{T}_G(C, X^*)$ connected in $C - Y$.*

Proof. Let $\mathcal{C}'$ denote the set of all components $C \in \mathcal{C}$ for which $Y \cap V(C)$ misses a vertex in $\mathcal{S}_G(C, X^*)$ or leaves a pair of vertices in $\mathcal{T}_G(C, X^*)$ connected in $C - Y$. Observe that for none of these components C, the graph $C - Y$ contains a path on at least two vertices where both endpoints are adjacent to the same vertex $u \in X^*$ in $G - Y$, as this contradicts Y being a feedback vertex set. Therefore for each $C \in \mathcal{C}'$, there exists a path in $G - Y$ containing a vertex in C such that both endpoints are distinct vertices from X^*. By applying Proposition 3, we now obtain the required bound on $\mathcal{C}'$. □

We will also use the following lemma on the time complexity of determining minimum feedback vertex sets in a graph of bounded elimination distance to a forest. We should note that both results can also be derived from Courcelle's theorem as explained by Jansen and Pieterse in the context of graphs of bounded treedepth [32].

Lemma 2. *Let G be a graph and η an integer such that $\mathrm{ED}_{\mathcal{G}_F}(G) \leq \eta$. Let $S, T \subseteq V(G)$. Then one can compute $\mathrm{FVS}(G)$ in time $n^{\mathcal{O}_\eta(1)}$. Furthermore, one can determine whether there exists a minimum feedback vertex set in G that contains S and separates T in time $n^{\mathcal{O}_\eta(1)}$.*

We remark that an even stronger statement is true: the quantities can be computed in time $f(\eta) \cdot n^{\mathcal{O}(1)}$ using extensions of Courcelle's theorem and the fact that the treewidth of a graph is bounded in terms of its elimination distance to a forest; see Lemma 5 in [32]. This improvement will not be relevant for us, though.

Proof sketch. As the approach is standard while working out the details would be tedious, we only sketch the high-level ideas for completeness. Observe that by Proposition 1. 1, G also has treewidth at most $\eta + 1$ and we can determine a tree decomposition of width $\eta + 1$ in polynomial time for fixed η [3]. We let $\left(R, (B_u)_{u \in V(R)}\right)$ denote such a tree decomposition. The size of a minimum feedback vertex set can be obtained with dynamic programming over this tree decomposition. The subproblem for node $t \in V(R)$ with $Z \subseteq B_t$ and a partition $\mathcal{P}$ of $B_t - Z$ can be defined as the size of a minimum feedback vertex set Y in G_t, such that

- $Y \cap B_t$ equals Z.
- Two vertices of B_t are in the same set in $\mathcal{P}$ if and only if they are in the same connected component in $G_t - Y$.

Since $|B_t| \leq \eta + 2$, the number of partitions can be bounded by $(\eta+2)^{\eta+2}$, which will yield a polynomial running time for constant η. We refer to [17, Section 7.3.3] for a description of the recurrences for these type of problems.

With a minor modification, we can also derive the size of a minimum set of vertices Y in G such that Y is a feedback vertex set, Y contains S and Y separates T in G. Observe that the minimum size of such a set is equal to the size of a minimum feedback vertex set, if and only if there exists a minimum feedback vertex set that also contains S and separates T. After removing S from the graph, we use a similar dynamic programming approach over a tree decomposition. The subproblem for node $t \in V(R)$ with $Z \subseteq B_t$ and a partition $\mathcal{P}$ of $B_t - Z$ now also takes a boolean value for each class of the partition and asks for the minimum size of a feedback vertex set Y in G_t that separates $T \cap V(G_t)$ such that

- $Y \cap B_t$ equals Z.
- Two vertices of B_t are in the same set in $\mathcal{P}$ if and only if they are in the same connected component in $G_t - Y$.
- A connected component of $G_t - Y$ contains no vertices in T if the boolean value for the corresponding class of the partition is False.
- A connected component of $G_t - Y$ contains precisely one vertex in T if the boolean value for the corresponding class of the partition is True.

When merging two subtrees, we must ensure that a connected component contains only one vertex in T after merging. We leave the details of the recursion to the reader. □

We will explore a special case of covering/packing duality on trees regarding multiway cuts, for which we use the following definition.

Definition 6 (*T-Path*)**.** Let G be a graph and let $T \subseteq V(G)$ be a set of vertices. A *T-path* is a path in G whose endpoints are two distinct vertices in T.

If for a graph G with $T \subseteq V(G)$ there exists a collection of k vertex-disjoint T-paths, then any vertex multiway cut of T in G has size at least k. While the converse is not true in general, it does hold on a tree. We provide a proof of this folklore statement for completeness.

Proposition 5. *Let G be a tree and let $T \subseteq V(G)$ be a set of terminals. If $Z \subseteq V(G)$ is a minimum vertex multiway cut of T with size k, i.e. in $G - Z$ all vertices in $T \setminus Z$ are in different connected components, then there exists a collection of k vertex-disjoint T-paths.*

Proof. For a vertex v in a rooted tree G, we use G_v to denote the subgraph of G induced by all descendants of v. Pick a root r of the tree arbitrarily. We will assume that Z has the following property: for any $u \in Z$ (other than the root) with parent w, the set $(Z \setminus \{u\}) \cup \{w\}$ is not a vertex multiway cut of T. Observe that we can construct such a vertex multiway cut greedily from any minimum vertex multiway cut by repeatedly replacing a vertex by its parent when the result still separates T in G.

We will now use induction on k, the size of a minimum vertex multiway cut. If $k = 1$, then observe that $|T| \geq 2$ as otherwise the empty set suffices to separate T. Therefore, we can identify a T-path in the tree. Now suppose that $k > 1$ and suppose that the claim holds for any tree with a minimum vertex multiway cut of size smaller than k. Take a tree G that has a minimum vertex multiway cut Z of size k. Find a vertex $u \in Z$ such that $G_u \cap Z = \{u\}$, i.e. there are no other vertices from the vertex multiway cut in the subtree rooted at u. Observe that we can always find such a vertex that is not the root of the tree. Then we can identify a T-path between two terminals in G_u: replacing u by its parent does not yield a vertex multiway cut, so there are at least two terminals in G_u. Observe that the T-path is contained in G_u as well. Now let G' be the graph where G_u is removed from G. Observe that it is a tree and that a minimum vertex multiway cut has size $k-1$: the set $Z \setminus \{u\}$ is a valid vertex multiway cut on G' and if a smaller vertex multiway cut Z' would exist on G', then $Z' \cup \{u\}$ would have been a smaller vertex multiway cut on G. Therefore, we can apply the induction hypothesis to obtain a collection of $k-1$ vertex-disjoint T-paths in G'. Together with the T-path that we identified in G_u, we obtain a collection of k vertex-disjoint paths in G, proving the induction step. □

We record the following consequence for later use.

Proposition 6. *Let G be a tree and $T' \subseteq V(G)$. If T' cannot be separated with η vertices, then there exist $\eta+1$ vertex-disjoint paths whose endpoints are distinct vertices in T'.*

Proof. Let Z be a minimum vertex multiway cut of T' in G, so $|Z| \geq \eta+1$. By Proposition 5, there are $\eta+1$ vertex-disjoint paths whose endpoints are distinct vertices in T'. □

We have now collected all the ingredients to prove our main lemma.

3.2. Proof of the main lemma

We repeat the formulation of Lemma 1 as stated in the introduction.

Lemma 1. *Let G be a connected graph with disjoint vertex sets $S, T \subseteq V(G)$. Suppose that any minimum feedback vertex set X of G misses some vertex from S or leaves two vertices from T connected in $G - X$. Then there exist sets $S^* \subseteq S$ and $T^* \subseteq T$ whose sizes only depend on the elimination distance to a forest of G, such that any minimum feedback vertex set X of G misses some vertex from S^* or leaves two vertices from T^* connected in $G - X$.*

We can split up Lemma 1 into two parts. Lemma 3 will bound the number of vertices in the $\mathcal{G}_F$-elimination tree that contain a vertex in S or T. This part corresponds to the original treedepth formulation in [32, Lemma 3], but is significantly simplified for our restricted setting. Lemma 4 bounds the number of vertices in S and T in a bag of the elimination tree.

Lemma 3. *Let G be a connected graph with disjoint vertex sets $S, T \subseteq V(G)$. Let $\big(R, (B_u)_{u \in V(R)}\big)$ be a $\mathcal{G}_F$-elimination tree of G of height η. Suppose that any minimum feedback vertex set X of G misses a vertex from S or leaves two vertices from T connected. Then this also holds for some subsets $S^* \subseteq S$ and $T^* \subseteq T$, such that any vertex in the elimination tree has at most $3\eta \cdot 2^\eta$ children u for which G_u contains a vertex from S^* or T^*.*

Lemma 4. *Let G be a connected graph with disjoint vertex sets $S, T \subseteq V(G)$. Let $\big(R, (B_u)_{u \in V(R)}\big)$ be a $\mathcal{G}_F$-elimination tree of G of height η. Suppose that any minimum feedback vertex set X of G misses a vertex from S or leaves two vertices from T connected. Then this also holds for some subsets $S^* \subseteq S$ and $T^* \subseteq T$, such that for any leaf u in the elimination tree, the set B_u contains at most $\eta + 1$ vertices from S^* and at most $\mathcal{O}(\eta^2)$ from T^*.*

Before proving these lemmata, we show how they imply Lemma 1.

Proof of Lemma 1. Let G be a graph with vertex sets S and T as stated in Lemma 1 and let $\big(R, (B_u)_{u \in V(R)}\big)$ be a $\mathcal{G}_F$-elimination tree of G. We can first apply Lemma 3 to obtain sets $S' \subseteq S$ and $T' \subseteq T$ satisfying the guarantee of that lemma: for each vertex in the elimination tree R, we can bound the number of children u for which B_u contains a vertex in S' or T'. Now define subgraph R' of elimination tree R to be the induced subgraph of R by precisely those vertices $u \in V(R)$ for which G_u contains some vertex in S' or T'. If R' is the empty graph, then S' and T' are empty, thereby directly implying Lemma 1. Otherwise, R' is still a rooted tree with height at most η. By Lemma 3, each vertex in this subtree has at most $3\eta \cdot 2^\eta$ children. Since the height of the tree is at most η, there are at most $(3\eta \cdot 2^\eta)^\eta$ vertices in each layer of R', thereby bounding the number of vertices in $u \in V(R)$ for which B_u contains a vertex from $S' \cup T'$. However, for leaves u of the elimination tree, there is no bound on the number of vertices in $B_u \cap (S' \cup T')$ yet. By applying Lemma 4 now on graph G and sets S' and T', we obtain sets $S^* \subseteq S' \subseteq S$ and $T^* \subseteq T' \subseteq T$ such that for each vertex u in elimination tree R, the number of vertices in $B_u \cap (S^* \cup T^*)$ is bounded by a function of η. Furthermore, since no node of $V(R) \setminus V(R')$ contains vertices of $S^* \cup T^*$ in its bag, this implies that $|S^* \cup T^*|$ is bounded by a function of η alone. This concludes the proof of Lemma 1. □

We now prove Lemma 3 and Lemma 4.

Proof of Lemma 3. In analogy to the original formulation in [32], we call a vertex of $S \cup T$ a *labeled vertex*. When we remove a label from a vertex, we remove the vertex from S and T while the vertex remains in the graph. Suppose that we pick $S^* \subseteq S$ and $T^* \subseteq T$ such that no minimum feedback vertex set contains S^* and separates T^*, while the latter property does not hold for any other pair of sets S', T' with $S' \subseteq S^*$ and $T' \subseteq T^*$. We refer to this property as the *minimality* of S^* and T^*. We claim that for such minimal sets S^* and T^*, any vertex in the elimination tree R has at most $3\eta \cdot 2^\eta$ children u for which G_u contains a labeled vertex. We will also refer to the set T^* as the set of *terminals*.

Assume towards a contradiction that vertex $v \in V(R)$ has more than $3\eta \cdot 2^\eta$ child subtrees with labels. Let these children be $c_1, \dots, c_\ell$, so that for each child c_i of v with $i \in [\ell]$ we have $G_{c_i} \cap (S^* \cup T^*) \neq \emptyset$. By minimality of the pair S^* and T^*, the pair of sets $S^* \setminus G_{c_i}$ and $T^* \setminus G_{c_i}$ does not satisfy the stated property. Hence there exists a minimum feedback vertex set X_i in G which contains all vertices of $S^* \setminus G_{c_i}$ and separates $T^* \setminus G_{c_i}$. By assumption on S^* and T^*, we know that X_i misses a vertex in $S^* \cap G_{c_i}$ or leaves a vertex in $T^* \cap G_{c_i}$ connected to some other vertex in T^*. Based on the minimum feedback vertex set X_i witnessing the behavior of child c_i, we define $Z_i := \textsc{tail}[v] \setminus X_i$.

Now fix a set $Z \subseteq \textsc{tail}[v]$. We will bound the number of children c_i for which $Z_i = Z$ by 3η. Suppose towards a contradiction that there are $3\eta + 1$ of these children. Let C be the set containing these vertices. Fix a child $c_j \in C$ along with its witnessing feedback vertex set X_j, and observe the following.

- By Proposition 1. 4, there are at most η children $c_i \in C$ where $X_j \cap G_{c_i}$ is not a minimum feedback vertex set in G_{c_i}.
- There are at most η children $c_i \in C$ with $i \neq j$ such that a terminal in G_{c_i} is connected to a vertex in Z in $G^+_{c_i} - X_j$, i.e. (recall the notation in Section 2.1) in the induced subgraph on the remaining vertices in the subtree and tail of c_i. If there were $\eta + 1$ or more, by the pigeonhole principle there would be two children c_p, c_q other than c_j and respective terminals $t_p \in V(G_{c_p})$, $t_q \in V(G_{c_q})$, such that there is some vertex $z \in Z$ that both t_p and t_q can reach in the graph $G - X_j$; but this contradicts the assumption that X_j separates all terminals of $T^* \setminus G_{c_i}$ since we can connect $t_p, t_q \in T^* \setminus G_{c_i}$ by concatenating the two paths to z.

- There are at most $\eta - 1$ children $c_i \in C$ such that in $G_{c_i}^+ - X_j$, there exists a path between distinct vertices in Z that uses some vertex in G_{c_i}. Otherwise, we claim that we can directly construct a cycle in $G - X_j$. Consider for example the auxiliary (multi)graph on vertex set Z which contains, for each child $c_i \in C$ for which $G_{c_i}^+ - X_j$ contains such a path, say between $z_1, z_2 \in Z$, one edge z_1z_2. This auxiliary graph contains a cycle since it has too many edges to be acyclic, which implies that there exists a cycle in $G - X_j$.

Pick a child $c_k \in C$ which is neither c_j nor in the list of $3\eta - 1$ children above. As $|C| > 3\eta$, such a vertex exists. By the first item above, we can deduce that $X_j \cap G_{c_k}$ is a minimum feedback vertex set in G_{c_k}. Besides, this set contains $S^* \cap G_{c_k}$, it separates all terminals in $T^* \cap G_{c_k}$, and it separates $T^* \cap G_{c_k}$ from Z. Furthermore, no path exists in $G_{c_k}^+ - X_j$ that connects two vertices in Z and also contains some vertex in G_{c_k}.

Claim 1. *The set $X' := (X_k \setminus G_{c_k}) \cup (X_j \cap G_{c_k})$ is a minimum feedback vertex set in G which contains S^* and separates T^*.*

Proof. First, observe that any cycle that is not trivially broken must contain vertices both in G_{c_k} and outside G_{c_k}. However, when such a cycle leaves G_{c_k}, it must enter Z. This cycle cannot contain only one vertex in Z, as it would then also be a cycle in $G_{c_k}^+ - X_j$. As no pair of vertices in Z is connected by a path containing some vertex in G_{c_k}, such a cycle cannot exist at all. Since $X_j \cap G_{c_k}$ is a minimum feedback vertex set in G_{c_k}, it follows that X' must be a minimum feedback vertex set in G.

Furthermore, $X_j \cap G_{c_k}$ contains $S^* \cap G_{c_k}$ and separates $T^* \cap G_{c_k}$. As $X_k \setminus G_{c_k}$ does the same for the remainder of S^* and T^*, we only need to verify that no terminal in G_{c_k} is connected to a terminal outside G_{c_k}. Because $T^* \cap G_{c_k}$ is also separated from $\textsc{tail}[v]$, this cannot happen. Therefore, X' is a minimum feedback vertex set containing S^* and separating T^*. ■

Claim 1 contradicts that any minimum feedback vertex set in G misses a vertex in S^* or leaves two vertices in T^* connected. We conclude that there are at most 3η children c_i of v for which a witnessing minimum feedback vertex set has $Z_i = Z$. As there are at most 2^η subsets of $\textsc{tail}[v]$ for any non-leaf v, this leads to the bound of at most $3\eta \cdot 2^\eta$ children for which the labels cannot be removed. □

It remains to prove Lemma 4.

Proof of Lemma 4. Pick some leaf v of elimination tree R, for which we want to ensure that there are $\mathcal{O}(\eta^2)$ vertices with labels among vertices in $Y := B_v$. Define $S_Y := S \cap Y$ and $T_Y := T \cap Y$. Our goal is to obtain subsets $S_Y^* \subseteq S_Y$ and $T_Y^* \subseteq T_Y$ whose sizes are $\mathcal{O}(\eta^2)$, such that every minimum feedback vertex set misses a vertex from $S_Y^* \cup (S \setminus Y)$ or leaves a pair of terminals in $T_Y^* \cup (T \setminus Y)$ connected. By applying this operation to all leaves of the elimination tree, we obtain the sets promised by Lemma 4.

The construction of S_Y^* is straightforward. If $|S_Y| > \eta + 1$, we let S_Y^* be an arbitrary subset of S_Y of size $\eta + 1$. Otherwise, $S_Y^* = S_Y$.

Claim 2. *Let X be a minimum feedback vertex set in G. If X misses a vertex in S, then it also misses a vertex in $S_Y^* \cup (S \setminus Y)$.*

Proof. If X misses a vertex in $S \setminus Y$, then the implication is trivial. Therefore assume X misses a vertex in S_Y. If this vertex is not in S_Y^*, then $|S_Y^*| = \eta + 1$ by construction. By Proposition 1. 2, we know that $|X \cap S_Y^*| \leq \eta$ so X misses a vertex in S_Y^*. ■

For the construction of T_Y^* we distinguish two cases. First, we assume that T_Y cannot be separated with η vertices in $G[Y]$. We then define T_Y^* by taking the $2\eta + 2$ endpoints of the paths guaranteed by Proposition 6. Observe that these vertices are all in T_Y.

Claim 3. *Suppose that T_Y cannot be separated with η vertices in $G[Y]$. Let X be a minimum feedback vertex set in G. Then X leaves two vertices in T_Y^* connected.*

Proof. By Proposition 1. 2, X can only intersect η of the $\eta + 1$ vertex-disjoint paths that were obtained through Proposition 6. Therefore, at least one path is disjoint from X, so its endpoints in T_Y^* are connected in $G - X$. ■

It remains to consider the case where T_Y can be separated with η vertices. Let Z be a vertex multiway cut of T_Y in $G[Y]$ with $|Z| \leq \eta$ and let $\mathcal{C} := \mathrm{cc}(G[Y] - Z)$. Observe that each of these connected components is a tree with at most one vertex in T_Y. Let $\mathcal{C}_T \subseteq \mathcal{C}$ be the set of components that contain a vertex in T_Y. We are now going to mark components. For each $z \in Z$, mark $\eta + 2$ components in $\mathcal{C}_T$ that are adjacent to z in $G[Y]$, or all if there are fewer. Similarly, for each $u \in \textsc{tail}(v)$ we mark up to $\eta + 2$ components in $\mathcal{C}_T$ that are adjacent to v in G_v^+. Then we define T_Y^* to be the union of all vertices in T_Y in the marked components, together with $Z \cap T_Y$. These are at most $\eta(\eta + 2) + \eta(\eta + 2) + \eta = \mathcal{O}(\eta^2)$ vertices.

Claim 4. *Suppose that T_Y can be separated with η vertices in $G[Y]$. Let X be a minimum feedback vertex set in G and suppose that X leaves two vertices in T connected. Then X also leaves two vertices in $T_Y^* \cup (T \setminus Y)$ connected.*

Proof. Let Z be the vertex multiway cut used in the construction of T^*_Y and let $t_1, t_2 \in T$ be two terminals that are connected in $G - X$. If they are both in $T^*_Y \cup (T \setminus Y)$, then the implication is trivial, so assume that $t_1 \in T_Y \setminus T^*_Y$. Observe that therefore $t_1 \notin Z$. Let P be a path from t_1 to t_2 in $G - X$ and let z be the first vertex on this path that is not in $G[Y] - Z$. We now distinguish two cases. If $z \in Z$, then observe that t_1 was in a component in $\mathcal{C}_T$ that was not marked. Then there are $\eta + 2$ marked components in $\mathcal{C}_T$ adjacent to z in $G[Y]$ of which the terminals are in T^*_Y. Only η of these components can be intersected by X by Proposition 1. 2, so there exists a path between two terminals in T^*_Y in $G[Y]$. If $z \notin Z$, then we obtain that $z \in \textsc{tail}(v)$ by Proposition 1. 3 and the case follows analogously. ■

This concludes the construction of the sets S^*_Y and T^*_Y. If any minimum feedback vertex set in G misses a vertex in S or leaves a pair of terminals in T connected, then it also misses a vertex in $S^*_Y \cup (S \setminus Y)$ or leaves a pair of terminals in $T^*_Y \cup (T \setminus Y)$ connected. By applying this operation to all leaves of the elimination tree, we obtain the promised sets S^* and T^* which concludes the proof of Lemma 4. □

Having proven the main lemma, in the next section we show how it leads to the desired kernelization algorithm.

3.3. Kernelization algorithm

Our kernel follows the structure of the polynomial kernel for $\mathcal{F}$-Minor Free Deletion when parameterized by a treedepth-η modulator for some integer η [32]. Our kernel relies crucially on the reduction rule specified in Lemma 5, whose correctness relies on Lemma 1. The role of this lemma for our kernelization is inspired by the use of [32, Lemma 6] in earlier work.

Lemma 5. *There is a polynomial-time algorithm that, given a graph G with modulator $X \subseteq V(G)$ such that $\textsc{ed}_{\mathcal{G}_F}(G - X) \leq \eta$ for a constant η, outputs an induced subgraph G' of G together with an integer Δ such that $\textsc{fvs}(G) = \textsc{fvs}(G') + \Delta$ and $G' - X$ has at most $|X|^{\mathcal{O}_\eta(1)}$ components.*

We will prove the lemma at the end of the section. It can be used to obtain a graph G' where $G' - X$ has a bounded number of connected components. We can then identify a set of vertices $Y \subseteq V(G' - X)$ with $|Y| \leq |X|^{\mathcal{O}_\eta(1)}$ such that $\textsc{ed}_{\mathcal{G}_F}(G' - X - Y) < \eta$. By definition of elimination distance, every connected component C of $G' - X$ contains a vertex whose removal decreases $\textsc{ed}_{\mathcal{G}_F}(C)$. As we limited the number of connected components by applying Lemma 5, these vertices constitute a suitable set Y. Now observe that $X \cup Y$ is a modulator to a graph with elimination distance to a forest $\eta - 1$ and that $|X \cup Y|$ is bounded by a polynomial in $|X|$. One can therefore provide an inductive argument which repeatedly applies Lemma 5 and increases the modulator such that the elimination distance to a forest of the remaining graph decreases every iteration. Once we obtain a modulator to a graph with elimination distance to a forest equal to zero (which is a forest), we can apply a known polynomial kernel in the size of a feedback vertex set [27]. We formalize these ideas in Theorem 3.

Theorem 3. *Let $\eta \geq 0$ be an integer. Let $\mathcal{G}_\eta$ be the set of graphs with elimination distance to a forest at most η. Then* Feedback Vertex Set *admits a polynomial kernel in the size of a $\mathcal{G}_\eta$-modulator X.*

For our kernel, we will assume that such a $\mathcal{G}_\eta$-modulator is provided along with the input. This allows us to decouple the complexity of finding a modulator from the complexity of exploiting its structure for kernelization, and bypasses a technical difficulty with structural parameterizations as described by Fellows et al. [21, §2.2]. In practice, one can determine a $\log(c + 1)$-approximation for the deletion distance to a minor-closed graph family with elimination distance to a forest bounded by c [25], since the treewidth is bounded as well by Proposition 1. 1. This approximation with an effectively constant approximation ratio improves the $|X_{\text{OPT}}| \log^{3/2} |X_{\text{OPT}}|$-approximation which was used in earlier papers [8,23]. Here, X_{OPT} denotes a minimum-sized modulator. Using this approximation algorithm, our kernelization algorithm can be used even when no modulator is known.

Proof of Theorem 3. Recall that X is a subset of $V(G)$ such that $\textsc{ed}_{\mathcal{G}_F}(G - X) \leq \eta$. We will prove the theorem using induction on η.

If $\eta = 0$ then $G - X$ is a forest. It is known that feedback vertex set admits a quadratic kernel in the size of a solution [17, Section 9.1], concluding the base case.

If $\eta > 0$, we apply Lemma 5 to obtain a graph G' and an integer Δ such that $G' - X$ has at most $|X|^{\mathcal{O}_\eta(1)}$ connected components and $\textsc{fvs}(G') + \Delta = \textsc{fvs}(G)$. Observe that G' and Δ can be obtained in polynomial time for fixed η. We will now extend the modulator X to a modulator X' such that $G' - X'$ has $\textsc{ed}_{\mathcal{G}_F}(G' - X') < \eta$. Consider each connected component C of $G' - X$ for which $\textsc{ed}_{\mathcal{G}_F}(C) = \eta$. By definition, for each such component C there must exist a vertex $v \in V(C)$ such that $\textsc{ed}_{\mathcal{G}_F}(C - \{v\}) < \eta$. We can find such a vertex by trying all options for v, computing $\textsc{ed}_{\mathcal{G}_F}(C - \{v\})$, and selecting a vertex v for which we find $\textsc{ed}_{\mathcal{G}_F}(C - \{v\}) < \eta$. This can be done in polynomial time for constant η since determining elimination distances to minor-closed graph classes is fixed-parameter tractable in the target value [11]. By adding this vertex to X for each considered component C, we obtain a set X' whose size is still bounded by a polynomial in $|X|$ for constant η since the number of components to consider is polynomial in $|X|$. Since each connected component C of $G' - X'$ has $\textsc{ed}_{\mathcal{G}_F}(C) < \eta$, it follows that $\textsc{ed}_{\mathcal{G}_F}(G' - X') < \eta$. Then by induction, the Feedback Vertex Set problem on G' admits a polynomial kernel in the size of X', which is a modulator to a graph with elimination distance to a forest at most $\eta - 1$. Since $\textsc{fvs}(G') + \Delta = \textsc{fvs}(G)$ by Lemma 5, we know that G' has a feedback vertex set of size at most ℓ if and only if G has a feedback vertex set of size at most $\ell + \Delta$. This concludes the induction step. □

It remains to prove Lemma 5.

Proof of Lemma 5. Pick $\gamma \in \mathcal{O}_\eta(1)$ such that for any graph with elimination distance to a forest at most η and vertex sets S and T, Lemma 1 returns subsets $S^* \subseteq S$ and $T^* \subseteq T$ of size at most γ. Define $\tau = |X| + 1 + 3\gamma$ and $\mathcal{C} = \text{cc}(G - X)$.

We are now going to add components from $\mathcal{C}$ to a set $\mathcal{C}'$. We will ensure that the size of $\mathcal{C}'$ is polynomial in $|X|$ for constant η and use this set to prove the lemma. Consider each set $X_0 \subseteq X$ with $|X_0| \leq 3\gamma$. Observe that these are $\mathcal{O}(|X|^{3\gamma}) = |X|^{\mathcal{O}_\eta(1)}$ subsets to consider. For each of these sets $X_0 \subseteq X$, we do the following for each component $C \in \mathcal{C}$: we consider the sets $\mathcal{S}(C, X_0)$ and $\mathcal{T}(C, X_0)$ as defined in Definition 5. We then determine whether there exists a minimum feedback vertex set in C containing $\mathcal{S}(C, X_0)$ and separating $\mathcal{T}(C, X_0)$. By Lemma 2, this can be verified in polynomial time in graphs with bounded elimination distance to a forest. If there are at most τ components in $\mathcal{C}$ where such a minimum feedback vertex set does not exist, then we add all of these components to $\mathcal{C}'$; otherwise we pick τ of them. Observe that $|\mathcal{C}'| = |X|^{\mathcal{O}_\eta(1)}$, since we add at most τ components for each considered subset of X.

We now claim the following regarding components that are not in $\mathcal{C}'$.

Claim 5. *Any component $C^* \in \mathcal{C} \setminus \mathcal{C}'$ satisfies $\textsc{fvs}(G) = \textsc{fvs}(G - V(C^*)) + \textsc{fvs}(C^*)$.*

Proof. Let $\widehat{G} = G - V(C^*)$. Let Y be a minimum feedback vertex set in G. Then Y can be partitioned into a feedback vertex set Y_1 in $\widehat{G}$ and a feedback vertex set Y_2 in C^*. It follows that $\textsc{fvs}(G) = |Y| = |Y_1| + |Y_2| \geq \textsc{fvs}(\widehat{G}) + \textsc{fvs}(C^*)$, and it remains to prove the other inequality.

Let $\widehat{Y}$ be a minimum feedback vertex set in $\widehat{G}$ and let $X_0 = X \setminus \widehat{Y}$. We again consider sets $\mathcal{S}(C^*, X_0)$ and $\mathcal{T}(C^*, X_0)$. If there exists a minimum feedback vertex set Y^* in C^* that contains $\mathcal{S}(C^*, X_0)$ and separates $\mathcal{T}(C^*, X_0)$, then by Proposition 2, $Y^* \cup \widehat{Y}$ is a minimum feedback vertex set in G. It follows that $\textsc{fvs}(G) \leq \textsc{fvs}(\widehat{G}) + \textsc{fvs}(C^*)$ under these circumstances. Therefore, suppose towards a contradiction that each minimum feedback vertex set in C^* misses a vertex in $\mathcal{S}(C^*, X_0)$ or leaves a pair of vertices in $\mathcal{T}(C^*, X_0)$ connected. Then by Lemma 1, this also holds for sets $S^* \subseteq \mathcal{S}(C^*, X_0)$ and $T^* \subseteq \mathcal{T}(C^*, X_0)$ whose sizes are bounded by γ.

Now consider the following set $X^* \subseteq X_0$: for each vertex in S^*, add two arbitrary neighbors in X_0 to X^*. For each vertex in T^*, add its neighbor in X_0 to X^*. Now consider the sets $\mathcal{S}(C^*, X^*)$ and $\mathcal{T}(C^*, X^*)$. Observe that $S^* \subseteq \mathcal{S}(C^*, X^*)$ and $T^* \subseteq \mathcal{T}(C^*, X^*)$. It follows that any minimum feedback vertex set in C^* misses a vertex in $\mathcal{S}(C^*, X^*)$ or leaves a pair of vertices in $\mathcal{T}(C^*, X^*)$ connected. Combined with the fact that $|X^*| \leq 3\gamma$, it means that C^* could have been added to $\mathcal{C}'$ during its construction when we considered X^* as a subset of X. However, the component was not added, implying that $\mathcal{C}'$ contains at least τ components C where every minimum feedback vertex set misses a vertex from $\mathcal{S}(C, X^*)$ or leaves a pair of vertices in $\mathcal{T}(C, X^*)$ connected. Let $\mathcal{C}''$ be the set of these components. By Proposition 4, there can be at most 3γ components $C \in \mathcal{C}''$ where $\widehat{Y} \cap V(C)$ misses a vertex from $\mathcal{S}(C, X^*)$ or leaves a pair of vertices in $\mathcal{T}(C, X^*)$ connected in $C - \widehat{Y}$. Since $\tau = |X| + 1 + 3\gamma$, there are at least $|X| + 1$ components $C \in \mathcal{C}''$ where $\widehat{Y} \cap V(C)$ contains $\mathcal{S}(C, X^*)$ and separates $\mathcal{T}(C, X^*)$, implying that $\widehat{Y} \cap V(C)$ is not a minimum feedback vertex set in C. We will now construct a different feedback vertex set $\widehat{Y}'$ on $\widehat{G}$:

- For each of the at least $|X| + 1$ components $C \in \mathcal{C} \setminus \{C^*\}$ where $\widehat{Y} \cap V(C)$ is not a minimum feedback vertex set of $G[C]$, we add a minimum feedback vertex set in C.
- For every other component $C \in \mathcal{C} \setminus \{C^*\}$, we take $\widehat{Y} \cap V(C)$.
- We add X to $\widehat{Y}'$.

Observe that $\widehat{Y}'$ is a feedback vertex set in $\widehat{G}$: since X is included in $\widehat{Y}'$, any cycle must be contained in a component in $\mathcal{C} \setminus \{C^*\}$, but for each of these components we added a feedback vertex set in C to $\widehat{Y}'$. Furthermore, $|\widehat{Y}'| \leq |\widehat{Y}| - (|X| + 1) + |X| < |\widehat{Y}|$, contradicting the assumption that $\widehat{Y}$ is a minimum feedback vertex set.

We conclude that there exists a minimum feedback vertex set in C^* that contains $\mathcal{S}(C^*, X_0)$ and separates $\mathcal{T}(C^*, X_0)$, proving that $\textsc{fvs}(G) \leq \textsc{fvs}(\widehat{G}) + \textsc{fvs}(C^*)$ as we saw earlier. ■

We can now complete the proof of Lemma 5. Let G' be the subgraph of G induced by X and all vertices in components in $\mathcal{C}'$. Observe that the number of components of $G' - X$ is polynomial in $|X|$. Let Δ be the size of a minimum feedback vertex set in the subgraph consisting of all components in $\mathcal{C} \setminus \mathcal{C}'$, which we can obtain in polynomial time by Lemma 2. By applying Claim 5 to all components in $\mathcal{C} \setminus \mathcal{C}'$, we conclude that $\textsc{fvs}(G) = \textsc{fvs}(G') + \Delta$. □

4. Kernelization lower bounds

In this section we focus on lower bounds. We first introduce some relevant terminology.

4.1. Preliminaries for kernelization lower bounds

We sometimes slightly abuse the $\mathcal{O}_\eta$ notation here and write $\mathcal{O}_\mathcal{F}(f(n))$ for a set of graphs $\mathcal{F}$, rather than writing the size of the largest graph in $\mathcal{F}$ as subscript. For a graph G and a set of connected graphs $\mathcal{F}$, we write $\textsc{ed}_{\neg\mathcal{F}}(G)$ to denote the elimination distance to an $\mathcal{F}$-minor free graph.

We introduce the notion of a necklace, which turns out to be a crucial structure.

Definition 7. Let G be a graph and let $\mathcal{F}$ be a collection of connected graphs. The graph G is an *$\mathcal{F}$-necklace* of length t if there exists a partition of $V(G)$ into $S_1, \ldots, S_t$ such that:

- $G[S_i] \in \mathcal{F}$ for each $i \in [t]$ (these subgraphs are the *beads* of the necklace),
- G has precisely one edge between S_i and S_{i+1} for each $i \in [t-1]$,
- G has no edges between any other pair of sets S_i and S_j.

When the length of the necklace is not relevant, we simply speak of an $\mathcal{F}$-necklace. The following definition specifies a special type of necklace.

Definition 8. Let $\mathcal{F}$ be a collection of connected graphs. Let G be an $\mathcal{F}$-necklace of length t. We say that G is a *uniform necklace* if it satisfies two additional conditions.

- There exists a graph $H \in \mathcal{F}$ such that each bead $G[S_i]$ is isomorphic to H.
- There exist $x, y \in V(H)$ and graph isomorphisms $f_i : V(H) \to V(G[S_i])$ for each bead $G[S_i]$, such that for each $i \in [t-1]$, the edge between $G[S_i]$ and $G[S_{i+1}]$ has precisely the endpoints $f_i(x)$ and $f_{i+1}(y)$.

Notice that for a collection of graphs $\mathcal{F}$, we can specify a uniform necklace uniquely by a graph $H \in \mathcal{F}$, its length t and the two vertices in $x, y \in V(H)$ indicating the endpoints of the edges between two consecutive beads. It will be useful to consider all uniform necklaces of different lengths with this structure. We speak of the *uniform necklace structure* (H, x, y) to describe all uniform $\{H\}$-necklaces where there exist isomorphisms as described in Definition 8: for each pair of consecutive beads S_i and S_{i+1} with edge e connecting them, the endpoint of e in S_i equals the image of x under the ith isomorphism and the endpoint in S_{i+1} equals the image of y under the $(i+1)$st.

The following proposition explains why we consider this special type of necklaces.

Proposition 7. *Let $\mathcal{F}$ be a finite collection of connected graphs. Let $\mathcal{G}$ be a minor-closed graph family that contains arbitrarily long $\mathcal{F}$-necklaces. Then $\mathcal{G}$ also contains arbitrarily long uniform $\mathcal{F}$-necklaces. Moreover, there exists a graph $H \in \mathcal{F}$ and vertices $x, y \in V(H)$ such that all uniform necklaces with structure (H, x, y) are contained in $\mathcal{G}$.*

Proof. We first argue that there exists a graph $H \in \mathcal{F}$ such that $\mathcal{G}$ contains arbitrarily long $\{H\}$-necklaces. Observe that if only a finite number c of beads in necklaces in $\mathcal{G}$ can be isomorphic to each graph in $\mathcal{F}$, then $\mathcal{G}$ cannot contain necklaces of length more than $c \cdot |\mathcal{F}|$. It follows that there exists a graph $H \in \mathcal{F}$ such that $\mathcal{G}$ contains necklaces where arbitrarily many beads are isomorphic to H. If we take such a necklace, contract all edges in beads which are not isomorphic to H and then contract the resulting paths between beads sufficiently, we can create arbitrarily long $\{H\}$-necklaces. These are in $\mathcal{G}$ as well since it is a minor-closed graph family. We can now take an $\{H\}$-necklace G of arbitrary length t in $\mathcal{G}$ with beads $G[S_1], \ldots, G[S_t]$. Let $f_i : V(H) \to S_i$ be a graph isomorphism between H and the ith bead for each $i \in [t]$. Now consider a bead $G[S_i]$ with $2 \leq i \leq t-1$ where $u \in S_i$ is the endpoint of the edge to S_{i-1} and $v \in S_i$ is the endpoint of the edge to S_{i+1}. There are $|V(H)|$ vertices in H that can be the preimage of u under the ith isomorphism and, similarly, there are $|V(H)|$ options for v. Therefore, by the pigeonhole principle there exist vertices $x, y \in V(H)$ such that for at least $\lceil \frac{t-2}{|V(H)|^2} \rceil$ beads $G[S_i]$, the vertex in S_i adjacent with S_{i-1} equals the image of x under the ith isomorphism and the vertex in S_i adjacent to S_{i+1} equals the image of y under that isomorphism. By contracting the edges within all other beads and by contracting the resulting paths between beads sufficiently, we obtain a uniform $\mathcal{F}$-necklace of length $\lceil \frac{t-2}{|V(H)|^2} \rceil$ which is contained in $\mathcal{G}$ as it is minor-closed. By picking t arbitrarily large, we conclude that $\mathcal{G}$ contains arbitrarily long uniform $\mathcal{F}$-necklaces.

Furthermore, suppose that for some graph H and vertices $x, y \in V(H)$, $\mathcal{G}$ does not contain all uniform $\mathcal{F}$-necklaces with structure (H, x, y). Since $\mathcal{G}$ is minor-closed, this implies that there exists a constant c bounding the length of all uniform $\mathcal{F}$-necklaces with structure (H, x, y) in $\mathcal{G}$. Now suppose towards a contradiction that for no $H \in \mathcal{F}$ and $x, y \in V(H)$, $\mathcal{G}$ contains all uniform $\mathcal{F}$-necklaces with structure (H, x, y). As the number of structures is finite for a finite collection of finite graphs, we can take the maximum of these constants. This constant then bounds the length of any uniform necklace in $\mathcal{G}$, while we proved earlier that $\mathcal{G}$ contains arbitrarily long uniform necklaces. We obtain a contradiction and thereby conclude that there exists an $H \in \mathcal{F}$ and $x, y \in V(H)$ such that $\mathcal{G}$ contains all uniform necklaces with structure (H, x, y). □

The following observation follows directly from the definitions of a graph minor.

Observation 1. *Let G be a graph and H be a connected graph. If ϕ is a minor model of H in G, then the graph $G[\phi(V(H))]$ is connected.*

We will often restrict ourselves to necklaces where each graph $H \in \mathcal{F}$ is *biconnected*, meaning that H is connected and for any $v \in V(H)$ the graph $H - \{v\}$ is still connected. A *biconnected component* of a graph G is a maximal biconnected subgraph of G, i.e. it is a biconnected subgraph of G which is not a proper subgraph of any other biconnected subgraph of G. The following proposition gives a necessary condition for a graph to contain a biconnected graph as a minor.

Proposition 8. *Let H be a biconnected graph and let G be a graph which contains H as a minor. Then for any minimal minor model ϕ of H in G, the graph $G[\phi(V(H))]$ is biconnected. Furthermore, the graph $G[\phi(V(H))]$ is a minor of a biconnected component of G.*

Proof. Suppose that G contains H as a minor and let ϕ be a minimal minor model of H in G. Assume towards a contradiction that $G[\phi(V(H))]$ is not biconnected. Let $v \in \phi(V(H))$ be a vertex such that $G[\phi(V(H)) \setminus \{v\}]$ is not connected. Let $u \in V(H)$ be the vertex such that $v \in \phi(u)$. Suppose that for some component C of the disconnected graph $G[\phi(V(H)) \setminus \{v\}]$, its vertex set $V(C)$ is included in $\phi(u)$. Observe that any edge of $G[\phi(V(H))]$ with precisely one endpoint in $V(C)$ will have v as its other endpoint, since C is a component of $G[\phi(V(H)) \setminus \{v\}]$. It follows that any edge with at least one endpoint in C is entirely contained in $\phi(u)$ in $G[\phi(V(H))]$. Now consider the minor model ϕ' of H in G, defined by

$$\phi'(w) = \begin{cases} \phi(w) & \text{if } w \neq u, \\ \phi(w) \setminus V(C) & \text{if } w = u. \end{cases}$$

Then observe that $\phi'(V(H))$ is still connected and for any edge xy in H, there still exists an edge between $\phi'(x)$ and $\phi'(y)$ in $G[\phi'(V(H))]$, since such an edge does not contain an endpoint in C. It follows that ϕ is not a minimal minor model of H in G in this case.

In the remainder of the proof, we may therefore assume that no component C of $G[\phi(V(H)) \setminus \{v\}]$ is contained in $\phi(u)$. We derive a contradiction by showing that $H - \{u\}$ is disconnected. To this end, consider the subgraph $G' = G[\phi(V(H))] - \phi(u)$. Observe that G' still contains multiple components, since no component of $G[\phi(V(H)) \setminus \{v\}]$ is contained in $\phi(u)$. Now for each $x, y \in V(H) \setminus \{u\}$ for which $xy \notin E(H)$, remove all edges between a vertex in $\phi(x)$ and a vertex in $\phi(y)$, and contract all edges between vertices in $\phi(x)$ for each $x \in V(H) \setminus \{u\}$. By definition of a minor model, we obtain the graph $H - \{u\}$. However, after contracting and removing edges in a disconnected graph, the remaining graph is still disconnected. Thereby $H - \{u\}$ is disconnected, a contradiction to the assumption that H is biconnected.

We conclude that for any minimal minor model ϕ of H in G, the graph $G[\phi(V(H))]$ is biconnected. It follows directly that $G[\phi(V(H))]$ is a subgraph (and thereby also a minor) of a biconnected component of G: if it is not a proper subgraph of any biconnected component of G, then it is not a proper subgraph of any biconnected subgraph of G. But then, by definition, $G[\phi(V(H))]$ is a biconnected component of G itself. □

Proposition 8 directly implies that if some graph G contains a biconnected graph H as a minor, then some biconnected component of G contains an H-minor. This provides us with a useful tool for arguing that a necklace does not contain some biconnected graph as a minor. To this end, observe that we can characterize the biconnected components of a necklace in the following way. Notice that we do not require the graphs in $\mathcal{F}$ to be biconnected here.

Observation 2. *Let $\mathcal{F}$ be a collection of connected graphs and let G be an $\mathcal{F}$-necklace. Then any biconnected component of G with at least three vertices is a subgraph of a bead.*

The observation follows from the fact that for any pair of beads, we can identify an edge that is used on any path connecting those two beads. Consequently, the endpoints of such an edge form cut vertices separating the two beads, so that no biconnected component on three or more vertices can intersect two beads. Proposition 8 and Observation 2 imply the following result.

Lemma 6. *Let $\mathcal{F}$ be a collection of connected graphs and let G be an $\mathcal{F}$-necklace. Let $\mathcal{H}$ be a collection of biconnected graphs on at least three vertices. If no graph in $\mathcal{F}$ contains an $\mathcal{H}$-minor, then G contains no $\mathcal{H}$-minor.*

Proof. Assume towards a contradiction that G contains an H-minor for some $H \in \mathcal{H}$. Then by Proposition 8, G has a biconnected component with an H-minor. Since H has at least three vertices, this biconnected component must have at least three vertices as well. By Observation 2, such a biconnected component is a subgraph of a bead of G. We conclude that G has a bead containing an H-minor, contradicting that no graph in $\mathcal{F}$ contains an $\mathcal{H}$-minor. □

Lemma 6 argues that it suffices to hit the $\mathcal{H}$-minors in each bead of a necklace to ensure that a necklace has no $\mathcal{H}$-minors at all. We will use these ideas when $\mathcal{F}$ is a collection of proper subgraphs of graphs in $\mathcal{H}$. Lemma 6 also implies the following corollary.

Corollary 1. *Let $\mathcal{F}$ be a collection of planar graphs and let G be an $\mathcal{F}$-necklace. Then G is also planar.*

Proof. By Kuratowski's celebrated theorem [36], a graph is planar if and only if it contains no K_5 and no $K_{3,3}$ minor. Observe that the set $\mathcal{H} = \{K_5, K_{3,3}\}$ is a set of biconnected graphs on at least three vertices. Since no graph in $\mathcal{F}$ contains an $\mathcal{H}$-minor by Kuratowski's theorem, also G contains no $\mathcal{H}$-minor by Lemma 6. We conclude that G is planar by Kuratowski's theorem. □

4.2. Proof setup

We repeat the lower bound we aim to prove here as stated in the introduction.

Theorem 2. *Let $\mathcal{G}$ be a minor-closed family of graphs and let $\mathcal{F}$ be a finite set of biconnected planar graphs on at least three vertices. If $\mathcal{G}$ has unbounded elimination distance to an $\mathcal{F}$-minor free graph, then $\mathcal{F}$-Minor Free Deletion does not admit a polynomial kernel in the size of a $\mathcal{G}$-modulator, unless* NP $\subseteq$ coNP/poly.

Notice that when we consider $\mathcal{F}$-Minor Free Deletion for some collection of graphs $\mathcal{F}$, we can assume that there do not exist distinct graphs $G, H \in \mathcal{F}$ such that G is a minor of H. Our proof consists of two parts. We first derive the following property, where we say that a set contains arbitrarily long necklaces if there does not exist a constant c such that each necklace in the set has length at most c.

Lemma 7. *Let $\mathcal{F}$ be a finite collection of connected planar graphs. Any minor-closed graph family $\mathcal{G}$ with unbounded elimination distance to an $\mathcal{F}$-minor free graph contains arbitrarily long uniform $\mathcal{F}$-necklaces.*

Then we will prove the following lemma by giving a reduction from CNF Satisfiability parameterized by the number of variables [19].

Lemma 8. *Let $\mathcal{F}$ be a finite set of biconnected planar graphs on at least three vertices and let $\mathcal{G}$ be a minor-closed graph family. If $\mathcal{G}$ contains arbitrarily long uniform $\mathcal{F}$-necklaces, then $\mathcal{F}$-Minor Free Deletion does not admit a polynomial kernel in the size of a $\mathcal{G}$-modulator, unless* NP $\subseteq$ coNP/poly.

Lemma 7 and Lemma 8 together directly imply Theorem 2.

Proof of Theorem 2. By Lemma 7, graph family $\mathcal{G}$ contains arbitrarily long uniform $\mathcal{F}$-necklaces. We can therefore apply Lemma 8 to conclude that $\mathcal{F}$-Minor Free Deletion does not admit a polynomial kernel in the size of a $\mathcal{G}$-modulator, assuming NP $\not\subseteq$ coNP/poly. $\square$

4.3. Characterizing graph families with unbounded elimination distance to an $\mathcal{F}$-minor free graph

We will prove Lemma 7 here. Our proof follows the proof by Bougeret et al. when they characterize graph families with unbounded bridge-depth [8]. Similar to their work, for a family of connected graphs $\mathcal{F}$ we define $\textsc{nm}_{\mathcal{F}}(G)$ to be the length of a longest $\mathcal{F}$-necklace that a graph G contains as a minor. Our goal is now to prove the existence of a small set X such that $\textsc{nm}_{\mathcal{F}}(G - X) < \textsc{nm}_{\mathcal{F}}(G)$ as described in Lemma 9.

Lemma 9. *Let $\mathcal{F}$ be a collection of connected planar graphs. Then there exists a polynomial function $f_{\mathcal{F}} : \mathbb{N} \to \mathbb{N}$ such that for any connected graph G with $\textsc{nm}_{\mathcal{F}}(G) = t$, there exists a set $X \subseteq V(G)$ with $|X| \le f_{\mathcal{F}}(t)$ such that $\textsc{nm}_{\mathcal{F}}(G - X) < t$.*

Bougeret et al. showed that one can derive a bounding function when the considered structures satisfy the Erdős–Pósa property [8]. This also is the case for $\mathcal{F}$-necklaces when the graphs in $\mathcal{F}$ are connected and planar, so this approach would be suitable for our purposes as well. To derive a polynomial bound on the size of X, we use a different argument that uses treewidth and grid minors. Before proving Lemma 9, we present some intermediate steps. We utilize the well-known fact that any planar graph is a minor of a sufficiently large grid. We provide a proof below to obtain explicit constants on the grid size.

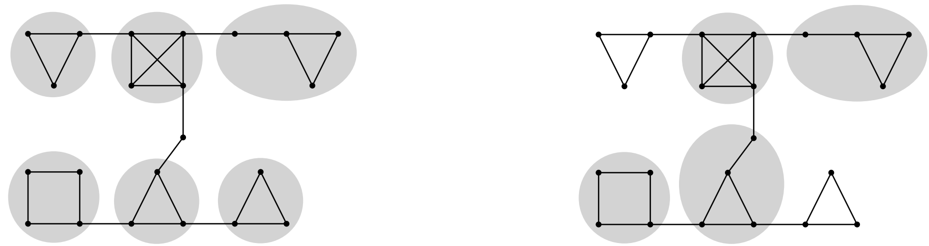
Proposition 9. *Any planar graph G on n vertices is a minor of the $4n \times 4n$ grid.*

Proof. We will first modify G by splitting vertices of large degree. We do this to ensure that each vertex in the modified graph has degree at most 4. For each vertex $u \in V(G)$ with degree $d(u) > 4$, we replace u by a path on $\lfloor d(u)/2 \rfloor$ vertices. Then we connect each neighbor of u to a vertex on this path, such that the maximum degree of each vertex on this path is 4. Observe that this is possible since both endpoints of the path can be connected to three neighbors, while the other vertices can be connected to two neighbors. Furthermore, notice that we can execute this procedure while ensuring that the modified graph is planar as well. We let G' denote the modified graph where each vertex has degree at most 4; observe that G is a minor of this graph.

As a planar graph on n vertices has at most $3n$ edges, the total degree of G is bounded by $6n$. We can use this to bound the number of vertices in G', since

$$|V(G')| \le \sum_{\substack{u \in V(G): \\ d(u) \le 4}} 1 + \sum_{\substack{u \in V(G): \\ d(u) > 4}} \lfloor d(u)/2 \rfloor \le n + \sum_{u \in V(G)} d(u)/2 \le 4n.$$

Any planar graph on n vertices that each have degree at most 4 can be drawn on an $n \times n$ grid such that the edges are non-intersecting grid paths [2]. It follows that there exists such an embedding on a grid with side length $4n$ for G'. Notice that G' is also a minor of this grid, so we conclude that also G is a minor. $\square$



(a) Two minor models of necklaces of length three. (b) A minor model of a necklace of length four.

Fig. 1. Fig. 1(a) displays a connected graph where two disjoint minor models of a $\{K_3\}$-necklace of length three are highlighted. Fig. 1(b) indicates how these necklaces can be combined into a necklace of length 4.

Together with the Excluded Grid Theorem, this proposition leads to the following treewidth bound.

Lemma 10. *Let $\mathcal{F}$ be a collection of connected planar graphs of at most n vertices each. There exists a polynomial $f : \mathbb{N} \to \mathbb{N}$ with $f(g) = \mathcal{O}(g^9 \text{poly} \log g)$ such that for any graph G with $\text{NM}_{\mathcal{F}}(G) = t$, it holds that $\text{TW}(G) < f(4n(t + 1))$.*

Proof. Let f be the function guaranteed by the Excluded Grid Theorem such that any graph with treewidth at least $f(g)$ contains the $(g \times g)$-grid as a minor [16]. Observe that $f(g) = \mathcal{O}(g^9 \text{poly} \log g)$. Now suppose towards a contradiction that the treewidth of G is at least $f(4n(t + 1))$. Then by the Excluded Grid Theorem, G contains the grid with side length $4(n(t + 1))$ as a minor. Now observe that an $\mathcal{F}$-necklace of length $t + 1$ has at most $n(t + 1)$ vertices when each graph in $\mathcal{F}$ has at most n vertices and observe that, by Corollary 1, $\mathcal{F}$-necklaces are planar when the graphs in $\mathcal{F}$ are planar. By Proposition 9, such an $\mathcal{F}$-necklace is therefore a minor of the grid with side length $4(n(t + 1))$. As G contains this grid as a minor, G also contains an $\mathcal{F}$-necklace of length $t + 1$ as a minor, contradicting that $\text{NM}_{\mathcal{F}}(G) = t$. We conclude that $\text{TW}(G) < f(4n(t + 1))$. □

To use this treewidth bound, we need a property similar to [8, Lemma 4.6].

Proposition 10. *For any family of connected graphs $\mathcal{F}$ and any connected graph G with $\text{NM}_{\mathcal{F}}(G) > 0$, any pair of minor models of $\mathcal{F}$-necklaces of length $\text{NM}_{\mathcal{F}}(G)$ in G must intersect.*

Proof. Fig. 1 illustrates the main idea of the proof. Define $t := \text{NM}_{\mathcal{F}}(G)$. Assume towards a contradiction that two disjoint minor models of an $\mathcal{F}$-necklace of length t exist. Then there exist pairwise disjoint sets $S_1, \dots, S_t, T_1, \dots, T_t$, such that:

- For each $i \in [t]$, both $G[S_i]$ and $G[T_i]$ are connected and they both contain an $\mathcal{F}$-minor.
- For each $i \in [t - 1]$, there exists an edge between S_i and S_{i+1}, as well as an edge between T_i and T_{i+1}.

Fig. 1 illustrates this setting and gives some intuition on how two of those necklaces can be combined into a longer one. Define $S := \bigcup_{i\in[t]} S_i$, $\mathcal{S} = \{S_1, \dots, S_t\}$ and define T and $\mathcal{T}$ analogously. Since G is connected, there exists a path between S and T. Consider a minimal such path P, so in particular only its endpoints are in $S \cup T$. Let S_i be the set in $\mathcal{S}$ containing the endpoint of P in S and let T_j be the set in $\mathcal{T}$ containing the endpoint of P in T. We will now construct an $\mathcal{F}$-necklace of length at least $t + 1$, contradicting that $\text{NM}_{\mathcal{F}}(G) = t$.

If $i \geq \frac{t+1}{2}$, then start the new $\mathcal{F}$-necklace minor model with sets $S_1, \dots, S_{i-1}$. Otherwise, start with $S_t, \dots, S_{i+1}$. Observe that also in the latter case, we select at least $t-(i+1)+1 \geq t - \frac{t+1}{2} = \frac{t-1}{2}$ sets. Transform set S_i into a set S_i' by including all vertices in P except the endpoint in T_j. We now include S_i' in the necklace. Then if $j \geq \frac{t+1}{2}$, we conclude the $\mathcal{F}$-necklace with sets $T_j, \dots, T_1$; otherwise with $T_j, \dots, T_t$. Observe that these are at least $\frac{t+1}{2}$ additional sets. We will now argue that these sets satisfy the properties of a necklace.

- The sets are pairwise disjoint, since the sets in $\mathcal{S} \cup \mathcal{T}$ are pairwise disjoint and since only one set was modified by adding vertices not occurring in $\mathcal{S} \cup \mathcal{T}$.
- For each vertex set, the subgraph of G induced by these vertices contains an $\mathcal{F}$-minor and is connected, since each set is either in $\mathcal{S} \cup \mathcal{T}$, or was in $\mathcal{S} \cup \mathcal{T}$ before adding vertices on a path with an endpoint in the set.
- Between any pair of consecutive sets, there exists an edge: the only non-trivial pair of consecutive sets is formed by S_i' and set T_j. By construction, there exists an edge in P which has one endpoint in S_i' and one in T_j, concluding this case as well.

We conclude that G contains an $\mathcal{F}$-necklace of length at least $\frac{t-1}{2} + 1 + \frac{t+1}{2} = t + 1$ as a minor, contradicting that $\text{NM}_{\mathcal{F}}(G) = t$. Therefore G does not contain two disjoint minor models of an $\mathcal{F}$-necklace of length $\text{NM}_{\mathcal{F}}(G)$. □

Proposition 10 is a generalization of the idea that in any connected graph, two paths of maximum length must intersect at a vertex. Given a graph G of treewidth w, we can use this property to identify at most $w + 1$ vertices (corresponding to a bag in an optimal tree decomposition) whose removal decreases $\text{NM}_{\mathcal{F}}(G)$. This result is described in Lemma 11.

Lemma 11. *Let $\mathcal{F}$ be a collection of connected graphs. Let G be a connected graph with $\text{TW}(G) = w$ and $\text{NM}_{\mathcal{F}}(G) = t$. Then there exists a set $Z \subseteq V(G)$ with $|Z| \leq w + 1$ such that $\text{NM}_{\mathcal{F}}(G - Z) < t$.*

Proof. Take a tree decomposition $\left(X, (B_u)_{u \in V(X)}\right)$ of G of width w. Now consider a node t of the decomposition such that G_t contains an $\mathcal{F}$-necklace of length t as a minor, but there is no child c of t for which the graph G_c contains such a minor. Observe that such a node exists, because for the root node r, the graph G_r contains such a minor. We now claim that B_t hits any minor model of an $\mathcal{F}$-necklace of length t in G.

First, notice that by definition of a tree decomposition, there cannot be an edge between a vertex u in $G_t - B_t$ and a vertex v in $G - V(G_t)$: if such an edge would exist, then the subtree of the tree decomposition of all bags containing u would intersect the subtree of all bags containing v, implying that B_t contains at least one of the vertices. It follows that any minor model of an $\mathcal{F}$-necklace of length t in $G - B_t$ must either be contained in $G_t - B_t$ or $G - V(G_t)$. The first set cannot contain such a minor model by construction of node t. The second set cannot either, as together with the necklace in G_t it would imply the existence of two disjoint minor models of an $\mathcal{F}$-necklace of length t in G, while Proposition 10 proves that any two of these models must intersect. We conclude that B_t is a set of at most $w + 1$ vertices hitting all minor models of $\mathcal{F}$-necklaces of length t in G. □

The proof of Lemma 9 follows directly by combining Lemma 10 and Lemma 11.

Proof of Lemma 9. By Lemma 10, we know that there exists a polynomial f such that $\text{TW}(G) < f(4n(t + 1))$. Then by Lemma 11, there exists a set X with $|X| \leq f(4n(t + 1))$ where n only depends on $\mathcal{F}$, as it is the maximum number of vertices in a graph in $\mathcal{F}$. □

We are almost in position to prove Lemma 7. We will first prove the following reformulation and then argue how it implies the lemma.

Lemma 12 (*Cf.* [8, *Theorem 4.8]*). *For any finite collection $\mathcal{F}$ of connected planar graphs, there exists a polynomial function g such that for any graph G,*

$$\text{ED}_{\neg\mathcal{F}}(G) \leq g(\text{NM}_{\mathcal{F}}(G)).$$

Proof. Let $\mathcal{F}$ be a finite collection of connected planar graphs. We will prove the statement by induction on $\text{NM}_{\mathcal{F}}(G) + |V(G)|$ for the function g defined by

$$g(t) = \sum_{i=1}^{t} f_{\mathcal{F}}(t),$$

where $f_{\mathcal{F}}$ denotes the function guaranteed by Lemma 9.

If $\text{NM}_{\mathcal{F}}(G) = 0$, then G does not contain any $\mathcal{F}$-minor, so $\text{ED}_{\neg\mathcal{F}}(G) = 0 = g(0)$.

Now suppose that $\text{NM}_{\mathcal{F}}(G) = t > 0$. We first consider the case where G is connected. Then by Lemma 9, there exists a set $X \subseteq V(G)$ with $|X| \leq f_{\mathcal{F}}(t)$ such that $\text{NM}_{\mathcal{F}}(G - X) < t$. Then for each component $C \in \text{cc}(G - X)$, we know that C is a connected graph with $\text{NM}_{\mathcal{F}}(C) < t$ and $|V(C)| \leq |V(G)|$, so by induction $\text{ED}_{\neg\mathcal{F}}(C) \leq g(\text{NM}_{\mathcal{F}}(C)) \leq g(\text{NM}_{\mathcal{F}}(G - X)) \leq g(t - 1)$. Since this holds for each component of $G - X$, we obtain by definition that $\text{ED}_{\neg\mathcal{F}}(G - X) = \max_{C \in \text{cc}(G-X)} \text{ED}_{\neg F}(C) \leq g(t - 1)$. From the definition of elimination distance, it follows that $\text{ED}_{\neg\mathcal{F}}(G) \leq |X| + \text{ED}_{\neg\mathcal{F}}(G - X)$. Intuitively, we can use $|X|$ rounds of the elimination process to eliminate all vertices of X, after which we are left with the graph $G - X$. This implies that

$$\text{ED}_{\neg\mathcal{F}}(G) \leq |X| + \text{ED}_{\neg\mathcal{F}}(G - X) \leq f_{\mathcal{F}}(t) + g(t - 1) = g(t),$$

which concludes the proof for the case that G is connected.

The case where G is disconnected is not much more difficult. For each component $C \in \text{cc}(G)$ we have $\text{NM}_{\mathcal{F}}(C) \leq \text{NM}_{\mathcal{F}}(G)$ while $|V(C)| < |V(G)|$, so by induction we can bound the elimination distance to an $\mathcal{F}$-minor free graph of C. As the elimination distance to an $\mathcal{F}$-minor free graph of G is the maximum of the elimination distance of its connected components, we thereby bound $\text{ED}_{\neg\mathcal{F}}(G)$ as well. □

We can now prove Lemma 7. Recall that for a finite collection $\mathcal{F}$ of connected planar graphs, Lemma 7 states that any minor-closed family of graphs $\mathcal{G}$ with unbounded elimination distance to an $\mathcal{F}$-minor free graph contains arbitrarily long uniform $\mathcal{F}$-necklaces. The proof of this lemma is fairly straightforward now.

Proof of Lemma 7. Let $\mathcal{G}$ be a minor-closed graph family containing graphs with arbitrarily large elimination distance to an $\mathcal{F}$-minor free graph. Suppose towards a contradiction that $\mathcal{G}$ contains no $\mathcal{F}$-necklace of length at least t for some t. Then no graph in $\mathcal{G}$ contains an $\mathcal{F}$-necklace of length t as a minor, thereby bounding the elimination distance to an $\mathcal{F}$-minor free graph of any graph in $\mathcal{G}$ by $g(t)$ by Lemma 12. We reach a contradiction and obtain that $\mathcal{G}$ contains arbitrarily long $\mathcal{F}$-necklaces, so by Proposition 7 we conclude that $\mathcal{G}$ also contains arbitrarily long uniform $\mathcal{F}$-necklaces. □

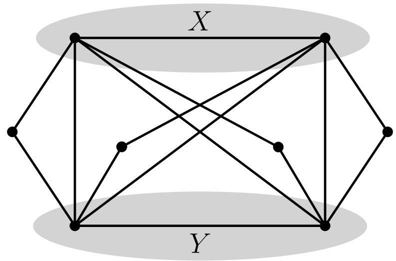


Fig. 2. Illustration of $J_{H,v,x,y}$ with $H = K_3$ and $v, x, y \in V(H)$. Observe that the subgraph induced by X is indeed isomorphic to $K_3 - \{v\}$.

4.4. Lower bound reduction

Our reduction is based on the following theorem on CNF Satisfiability. The theorem is a weaker form, in simplified terminology, of a result by Dell and van Melkebeek [19].

Theorem 4. *There is no polynomial-time algorithm that, given a CNF formula F on n variables with unbounded clause length, constructs an instance I of a fixed decision problem Q such that*

1. *F is satisfiable if and only if I is a* Yes*-instance of Q,*
2. *$|I| = \mathcal{O}(n^c)$ for some constant c,*

unless NP $\subseteq$ coNP/poly.

The restated form given above is easily seen to follow from [17, Thm. 15.29], since the existence of a reduction to instances of Q of size $\mathcal{O}(n^c)$ for formulas of arbitrary clause length, implies in particular that CNF Satisfiability with clauses of size $(c+1)$ has a compression of bitsize $\mathcal{O}(n^c)$, which yields NP $\subseteq$ coNP/poly.

We can use Theorem 4 to prove the non-existence of a polynomial kernel for $\mathcal{F}$-Minor Free Deletion parameterized by the size of a modulator to a graph with constant elimination distance to an $\mathcal{F}$-minor free graph. Our reduction will use the following graphs. An example is illustrated in Fig. 2.

Definition 9 ($J_{H,v,x,y}$)**.** Let H be a graph, $v, x, y \in V(H)$ with $x \neq y$ and let $c = |V(H)|$. Define $J_{H,v,x,y}$ to be the graph which can be constructed with the following two steps.

- We create $c-1$ vertices $x_1, \ldots, x_{c-1}$ for which we define the set $X := \{x_1, \ldots, x_{c-1}\}$. We then add precisely those edges such that $J_{H,v,x,y}[X]$ is isomorphic to $H - \{v\}$. We repeat this construction for a set of vertices $Y := \{y_1, \ldots, y_{c-1}\}$ such that $J_{H,v,x,y}[Y]$ is isomorphic to $H - \{v\}$.
- Then for any pair of $x_i \in X$ and $y_j \in Y$, we add $c-2$ extra vertices to the graph and let $D_{i,j}$ be the set containing these vertices. We then add edges to the graph such that $J_{H,v,x,y}[D_{i,j} \cup \{x_i, y_j\}]$ is isomorphic to H. In particular, there exists an isomorphism where x_i is the image of x and y_j is the image of y.

In total, the graph will have $2(c-1) + (c-1)^2(c-2)$ vertices.

The choice of vertices v, x and y will not be important, but they ensure that the graph is uniquely defined. The constructed graph has some useful properties.

Lemma 13. *Let H be a biconnected graph on at least three vertices and let $v, x, y \in V(H)$ with $x \neq y$. Let $J_{H,v,x,y}$ be the graph defined in Definition 9 with X, Y and c defined accordingly. Let $Z \subseteq V(J_{H,v,x,y})$ be a vertex set with $|Z| \leq c-1$. If $J_{H,v,x,y} - Z$ contains no H-minor, then Z is equal to X or Y.*

Proof. Assume towards a contradiction that Z is not equal to X or Y. We distinguish a few cases.

- If Z does not intersect $X \cup Y$, then Z cannot hit all H-minors which we added for each pair of vertices $x \in X$ and $y \in Y$: there are $(c-1)^2$ such pairs, giving a set of $(c-1)^2$ H-minors that only intersect each other in $X \cup Y$. As $(c-1)^2 > c-1 \geq |Z|$ for $c \geq 3$, the graph $J_{H,v,x,y} - Z$ contains an H-minor in this case.
- Suppose Z is contained in $X \cup Y$. Since Z is not equal to X and not equal to Y, there exist vertices $x_i \in X$ and $y_j \in Y$ such that $x_i, y_j \notin Z$. As Z contains no vertices outside $X \cup Y$, it cannot hit the H-minor that was constructed for this pair, so $J_{H,v,x,y} - Z$ contains an H-minor here as well.
- Now assume neither of those cases apply. Let $k = |Z \cap X|$ and $\ell = |Z \cap Y|$ and observe that $k + \ell \leq c-2$. Then the number of vertices in $X \setminus Z$ equals $c-1-k$ and the size of $Y \setminus Z$ equals $c-1-\ell$. It follows that the number of

pairs of a vertex in $X \setminus Z$ and one in $Y \setminus Z$ equals

$$\begin{aligned}(c-1-k)(c-1-\ell) &= (c-1)^2 - (c-1)(k+\ell) + k\ell \\ &\geq (c-1)^2 - (c-1)(c-2) \\ &= c-1.\end{aligned}$$

We now obtained at least $c-1$ H-minors where any pair of them can only intersect in a vertex in $(X \cup Y) \setminus Z$. Therefore, if these minors are all hit by Z, then Z needs to contain at least $c-1$ vertices outside $X \cup Y$. However, at least one vertex of Z is in $X \cup Y$, contradicting that $|Z| \leq c-1$.

We conclude that if Z is not X or Y, then $J_{H,v,x,y}$ contains an H-minor, thereby concluding the proof. □

The implication in Lemma 13 is actually an equivalence. For the other direction, we will prove a slightly stronger statement.

Lemma 14. *Let H be a biconnected graph on at least three vertices and let $v, x, y \in V(H)$ with $x \neq y$. Let $J_{H,v,x,y}$ be the graph defined in Definition 9 with X, Y and c defined accordingly. If $Z \subseteq V(J_{H,v,x,y})$ is equal to X or Y, then any biconnected graph that $J_{H,v,x,y} - Z$ contains as a minor is a proper minor of H.*

Proof. Assume without loss of generality that $Z = Y$. Now assume towards a contradiction that $J_{H,v,x,y} - Y$ contains an H'-minor for some biconnected H' that is not a proper minor of H. Observe that by Proposition 8, a biconnected component of $J_{H,v,x,y} - Y$ must contain an H'-minor. However, the only biconnected components of $J_{H,v,x,y} - Y$ are the induced subgraphs by the vertices $D_{i,j} \cup \{x_i\}$ for $i \in [c-1]$. These subgraphs are minors of H, so they cannot contain an H'-minor when H' is a proper minor of H. □

We can now give the reduction.

Lemma 15. *Let $\mathcal{F}$ be a finite collection of biconnected graphs on at least three vertices. Let $H \in \mathcal{F}$, let $x, y \in V(H)$ and consider the uniform necklace structure (H, x, y). There exists a polynomial-time algorithm that, given a CNF formula Ψ on n variables $x_1, \ldots, x_n$, outputs a graph G together with a set $X \subseteq V(G)$ of size $k = \mathcal{O}_{\mathcal{F}}(n)$ and an integer t, such that:*

1. *Ψ is satisfiable if and only if G has a solution to $\mathcal{F}$-Minor Free Deletion of size at most t, and*
2. *each connected component of $G - X$ is a minor of a uniform necklace with structure (H, x, y).*

Proof. We assume without loss of generality that for any $H \in \mathcal{F}$, no proper minor of H is contained in $\mathcal{F}$. If this were the case, we can remove H from $\mathcal{F}$ in the context of $\mathcal{F}$-Minor Free Deletion, as the deletion of all H-minors follows from the deletion of the proper minors of H.

We will start our construction of G by picking vertices $v, x, y \in V(H)$ with $x \neq y$ and let $c = |V(H)| = \mathcal{O}_{\mathcal{F}}(1)$. Given a CNF formula Ψ on variables $x_1, \ldots, x_n$, the algorithm executes the following three steps which are illustrated by Fig. 3:

1. For each variable x_i, we add a variable gadget J_i to G which is a copy of the graph $J_{H,v,x,y}$. Instead of using X and Y for the two subsets of $V(J_i)$ defined earlier, we use the names $V_{x_i} = \{v_{x_i,1}, \ldots, v_{x_i,c-1}\}$ and $V_{\neg x_i} = \{v_{\neg x_i,1}, \ldots, v_{\neg x_i,c-1}\}$. Notice the direct correspondence between literals and these sets.
2. For each clause in Ψ, we are going to add a necklace to G. For an integer t, we let N_t denote the uniform necklace with structure (H, x, y) of length t. For each clause in Ψ with length m, we add a copy of N_m to the graph. Each literal in this clause corresponds to a bead in the necklace. For $i \in [m]$, we let ℓ_i be the ith literal in this clause. We are now going to connect the ith bead to the vertices in V_{ℓ_i} for each $i \in [m]$. To do this, we pick a vertex u in the ith bead that is not adjacent to the other beads. As H has at least three vertices, such a vertex exists. We then connect u to V_{ℓ_i}, such that the subgraph induced by $V_{\ell_i} \cup \{u\}$ is isomorphic to H. By definition of $J_{H,v,x,y}$, we can ensure this by only adding edges between u and V_{ℓ_i}.
3. It remains to add one extra copy of H to G, whose vertex set we denote with S_0. We remove one edge from this copy of H, say with endpoints a and b. Then for every necklace that we added, we add an edge between a and the first bead, and an edge between b and the last bead of the necklace. We make sure that the endpoints in the necklace are not connected to a variable gadget.

Observe that, for fixed $\mathcal{F}$, the entire procedure can be executed in time polynomial in the size of Ψ.

Let $X := \left(\bigcup_{i \in [n]} (V_{x_i} \cup V_{\neg x_i})\right) \cup S_0$. Observe that $|X| = 2n(c-1) + c = \mathcal{O}_{\mathcal{F}}(n)$. We are now interested in the structure of the connected components of $G - X$. For each clause in Ψ, we added a uniform necklace with structure (H, x, y) in the second step. Since the neighbors of this necklace in G are all in X, these necklaces are connected components of $G - X$. The other connected components of $G - X$ are contained in the variable gadgets which were added in the first step. Each of these is a copy of H where two vertices are removed, so these components are minors of H. Therefore, each connected component of $G - X$ is a minor of a uniform necklace with structure (H, x, y).

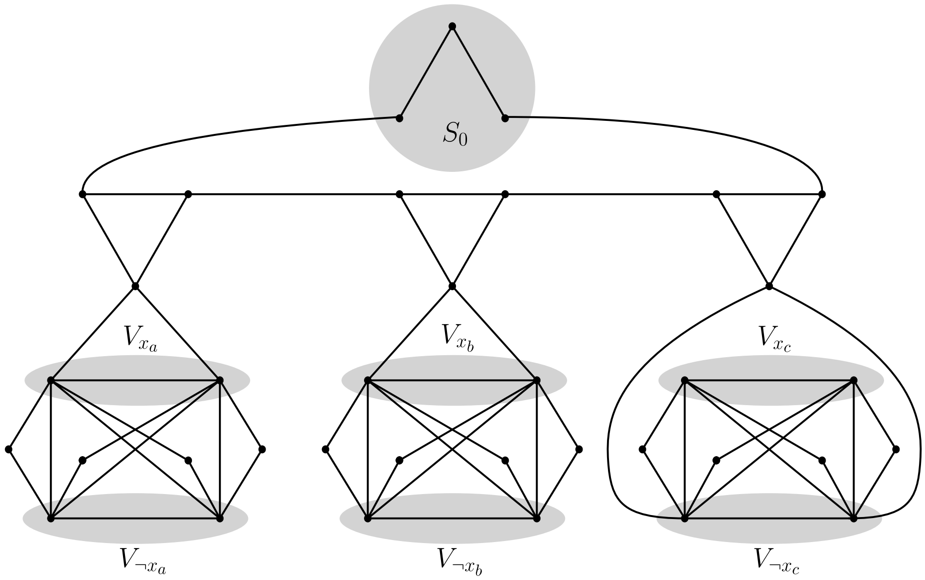


Fig. 3. Illustration of the reduction in the proof of Lemma 15 for a clause $(x_a \vee x_b \vee \neg x_c)$. Let ℓ_i denote the ith literal of this clause. We pick $H = K_3$ and we pick distinct $x, y \in V(H)$, so the uniform necklace structure becomes (K_3, x, y). We add such a uniform necklace of length three. The first and last bead are connected to S_0. We then connect each bead to a variable gadget. For the ith bead, we pick a vertex u in this bead which is not adjacent to the next or previous bead, and neither to S_0. We then ensure that the subgraph induced by u and V_{ℓ_i} equals K_3. Observe that, since the last literal of the clause is a negated variable, the last bead is connected to $V_{\neg x_c}$.

Let w be the total number of occurrences of literals in Ψ, i.e.

$$w := \sum_{\text{clause } C \text{ in } \Psi} \#\text{literals in } C.$$

We claim that Ψ is satisfiable if and only if G has a solution to $\mathcal{F}$-Minor Free Deletion of size at most $w + (c-1)n$. We will first prove the 'only if' direction.

Claim 6. *If Ψ is satisfiable, then G has a solution to $\mathcal{F}$-*Minor Free Deletion *of size at most $w + (c-1)n$.*

Proof. Suppose that Ψ has a satisfying truth assignment. Based on this truth assignment, we will construct a solution Z for $\mathcal{F}$-Minor Free Deletion of size $w + (c-1)n$ in the corresponding graph G. For every variable x_i that is set to True, we add the $c-1$ vertices in V_{x_i} to Z. For every variable set to False, we add $V_{\neg x_i}$ to Z. These vertices are in the variable gadgets added in the first step. Now, Z contains $(c-1)n$ vertices.

Then for each clause in Ψ, we consider each literal ℓ_i in this clause and consider the necklace in G corresponding to this clause. If ℓ_i evaluates to True according to the truth assignment, we add a vertex from the ith bead to Z that is adjacent to the previous or next bead. Otherwise, we add the vertex in this bead adjacent to V_{ℓ_i}. This procedure adds another w vertices to Z, ensuring that Z has size $w + (c-1)n$.

We will now prove that $G - Z$ contains no $\mathcal{F}$-minors. The graph $G - Z$ contains multiple connected components. We will argue that none of these components contains an $\mathcal{F}$-minor, as it implies directly that $G - Z$ has no $\mathcal{F}$-minors.

Consider a variable gadget J_i corresponding to some variable x_i, and assume without loss of generality (by symmetry) that x_i evaluates to True in the given truth assignment. The only vertices in J_i that were connected to other vertices in G are those in V_{x_i} and those in $V_{\neg x_i}$. The first set is removed. For the second set, observe that these vertices are only connected to beads in necklaces where the corresponding literal is $\neg x_i$, and that precisely these vertices in those beads are removed as $\neg x_i$ evaluates to False. It follows that $J_i - V_{x_i}$ is a connected component in $G - Z$. By Lemma 14, any biconnected minor of $J_i - V_{x_i}$ is a proper minor of H. By assumption, for each pair of graphs $H_1, H_2 \in \mathcal{F}$, H_1 is not a minor of H_2 and H_2 is not a minor of H_1. Since all graphs in $\mathcal{F}$ are biconnected, it follows that the connected component $J_i - V_{x_i}$ contains no $\mathcal{F}$-minors.

After analyzing the connected components containing vertices of variable gadgets, it remains to analyze the other connected components of $G - Z$. Define the set $\mathcal{F}'$ to be the set of all graphs that can be obtained by either removing a vertex or by removing an edge from a graph in $\mathcal{F}$. Observe that $(G - Z)[S_0] \in \mathcal{F}'$, and similarly the subgraph induced by what remains of a bead of a necklace is in $\mathcal{F}'$. Besides, since at least one literal evaluates to True in each clause, we always remove at least one edge between some pair of consecutive beads in a necklace. Hence, each necklace in G is now broken into at least two $\mathcal{F}'$-necklaces in $G - Z$. This also implies that in each necklace there does not exist a path through the necklace from the first to the last bead. Therefore, there exists at most one edge between each of those $\mathcal{F}'$-necklaces and vertex set S_0. Now consider the connected component that includes S_0. It follows that each biconnected component of at least three vertices of this component is either included in a bead or in S_0. In both cases, it is a minor of a graph in $\mathcal{F}'$. Observe that no graph in $\mathcal{F}'$ contains an $\mathcal{F}$-minor: then this graph would be a proper minor of H since $\mathcal{F}'$ contains only proper subgraphs of H. It follows that no biconnected component on at least three vertices contains an $\mathcal{F}$-minor, so this component contains no $\mathcal{F}$-minor by Proposition 8. It remains to observe that all other components of $G - Z$ are $\mathcal{F}'$-necklaces which are not connected to any other vertices. By Lemma 6, these components contain no $\mathcal{F}$-minors either. Since no component of $G - Z$ contains an $\mathcal{F}$-minor, $G - Z$ has no F-minors. ■

It remains to prove the opposite implication.

Claim 7. *If G has a solution to $\mathcal{F}$-Minor Free Deletion of size at most $w + (c - 1)n$, then Ψ is satisfiable.*

Proof. To this end, we will first observe that there are $w + (c - 1)n$ pairwise disjoint $\mathcal{F}$-minors in G. Each literal contributes such an H-minor in the bead in the corresponding $\{H\}$-necklace, giving w pairwise disjoint $\mathcal{F}$-minors. Now consider a variable gadget J_i corresponding to variable x_i. It contains the vertex sets $V_{x_i} = \{v_{x_i,1}, \ldots, v_{x_i,c-1}\}$ and $V_{\neg x_i} = \{v_{\neg x_i,1}, \ldots, v_{\neg x_i,c-1}\}$. For each pair of a vertex in V_{x_i} and a vertex in $V_{\neg x_i}$, we created an H-minor in this gadget. We can identify $c - 1$ pairwise disjoint H-minors in G by considering for each $j \in [c - 1]$ the H-minor created for the pair of $v_{x_i,j}$ and $v_{\neg x_i,j}$. Observe that these minors are pairwise disjoint and contained in J_i. Therefore each variable gadget contributes $c - 1$ H-minors, giving a total of $w + (c - 1)n$ H-minors which are all pairwise disjoint.

Consider a variable gadget J_i for variable x_i. If there exists a solution to $\mathcal{F}$-Minor Free Deletion within this variable gadget of at most $c - 1$ vertices, then by Lemma 13 the solution is equal to V_{x_i} or $V_{\neg x_i}$. Now consider a solution Z for $\mathcal{F}$-minor deletion in G of at most $w + (c - 1)n$ vertices. Since we can identify $w + (c - 1)n$ pairwise disjoint H-minors, Z contains one vertex from each of the identified H-minors in G. It follows that Z is disjoint from S_0 and contains exactly $c - 1$ vertices from each variable gadget. Then for a variable x_i, the intersection of Z with $V(J_i)$ equals precisely V_{x_i} or $V_{\neg x_i}$ by Lemma 13.

We can now construct a truth assignment for Ψ. For each variable x_i, we verify which vertices Z contains from J_i. If V_{x_i} is included (and therefore $V_{\neg x_i}$ is disjoint from Z), we set x_i to True, and if $V_{\neg x_i}$ is included we set x_i to False. It remains to prove that this yields a valid truth assignment.

To this end, we first consider an $\mathcal{F}$-necklace corresponding to a clause. Observe that Z contains precisely one vertex from each bead, as each bead contains one of the $w+(c-1)n$ pairwise disjoint H-minors. Suppose towards a contradiction that in each bead, Z contains the vertex which we connected to a variable gadget. Since this vertex is not adjacent to other beads by construction and since H is biconnected, it follows that there still exists a path from $a \in S_0$ through this necklace to $b \in S_0$. By contracting this path to a single edge, we obtain a copy of the graph H on vertex set S_0, contradicting that Z hits all H-minors. It follows that in each necklace corresponding to a clause, we can identify a bead where the vertex connected to the corresponding variable gadget is not in Z. Let this vertex be u in a bead corresponding to literal ℓ. We now claim that ℓ evaluates to True, implying that such a literal exists in every clause and thereby proving that Ψ is satisfiable. Assume towards a contradiction that ℓ evaluates to False, then in the corresponding variable gadget, none of the vertices of V_ℓ were included in Z. However, recall that we created an H-subgraph on vertex set $V_\ell \cup \{v\}$. As none of these vertices is in Z, this subgraph contradicts Z being a solution to $\mathcal{F}$-Minor Free Deletion. It follows that ℓ evaluates to True, so every clause in Ψ contains a literal that evaluates to True. This concludes the claim that Ψ is satisfiable. ■

These claims prove the correctness of our reduction and conclude the proof of Lemma 15. □

We are finally able to prove Lemma 8. We formalize this lemma with the following theorem and prove the theorem.

Theorem 5 (*Formalization of Lemma 8*). *Let $\mathcal{F}$ be a finite collection of biconnected graphs on at least three vertices. Let $\mathcal{G}$ be a minor-closed graph family containing arbitrarily long uniform $\mathcal{F}$-necklaces. There is no polynomial-time algorithm that, given a graph G, integer t and vertex set X of size k such that $G - X$ is in $\mathcal{G}$, constructs an instance I of a fixed decision problem Q such that:*

1. *G has a solution to $\mathcal{F}$-Minor Free Deletion of size at most t if and only if I is a Yes-instance of Q, and*
2. *$|I| = \mathcal{O}(k^c)$ for some constant c,*

unless NP ⊆ coNP/poly.

Proof. We will prove this theorem by contradiction: if such an algorithm would exist, we can use it to give a compression algorithm for CNF Satisfiability that would contradict Theorem 4. This can be achieved by reducing a provided CNF formula to an instance of $\mathcal{F}$-Minor Free Deletion.

Let Ψ be a CNF formula on n variables. By Proposition 7, there exists a uniform necklace structure (H, x, y) with $H \in \mathcal{F}$ and $x, y \in V(H)$ such that all uniform necklaces with this structure are contained in $\mathcal{G}$. For this uniform necklace structure (H, x, y), let G be the graph obtained by Lemma 15 with a set X of size $k = \mathcal{O}_{\mathcal{F}}(n)$ such that all connected components of $G - X$ are minors of uniform necklaces with this structure. Then observe that $G - X$ itself is also a minor of a sufficiently long uniform necklace with the same structure. Let t be the integer such that Ψ is satisfiable if and only if G has a solution to $\mathcal{F}$-Minor Free Deletion of size at most t.

If a compression algorithm as stated in the theorem would exist, we can construct an instance I of a fixed decision problem Q such that G has a solution to $\mathcal{F}$-Minor Free Deletion of size at most t if and only if I is a Yes-instance of Q, while $|I| = \mathcal{O}(k^c)$. Since $k = \mathcal{O}_{\mathcal{F}}(n)$, the bound on the size of I becomes $|I| = \mathcal{O}(n^c)$ when $\mathcal{F}$ is a finite collection of graphs. However, since G has a solution to $\mathcal{F}$-Minor Free Deletion of size at most t if and only if Ψ is satisfiable, we obtain a contradiction with Theorem 4. Therefore such an algorithm cannot exist, concluding the proof of Theorem 5. □

4.5. Proof of Theorem 1

By combining our previous results, we can now prove Theorem 1.

Theorem 1. *Let $\mathcal{G}$ be a minor-closed graph family and assume* NP $\not\subseteq$ coNP/poly. *Then* Feedback Vertex Set *admits a polynomial kernel in the size of a $\mathcal{G}$-modulator if and only if $\mathcal{G}$ has bounded elimination distance to a forest.*

Proof. Let $\mathcal{G}_\eta$ be the set of graphs with elimination distance to a forest at most η. Then by Theorem 3, Feedback Vertex Set admits a polynomial kernel in the size of a $\mathcal{G}_\eta$-modulator X. Now let $\mathcal{H}$ be a family of graphs where each graph has elimination distance to a forest at most η. Then Theorem 3 implies directly that Feedback Vertex Set also admits a polynomial kernel in an $\mathcal{H}$-modulator, since such a modulator is always also an $\mathcal{G}_\eta$-modulator since $\mathcal{H} \subseteq \mathcal{G}_\eta$.

Regarding the lower bound, Theorem 2 states that for any minor-closed graph family $\mathcal{G}$ and any finite set $\mathcal{F}$ of biconnected planar graphs on at least three vertices, if $\mathcal{G}$ has unbounded elimination distance to an $\mathcal{F}$-minor free graph then $\mathcal{F}$-Minor Free Deletion does not admit a polynomial kernel in the size of a $\mathcal{G}$-modulator, unless NP $\subseteq$ coNP/poly. By taking $\mathcal{F} = \{K_3\}$, we obtain the stated lower bound. □

5. Conclusion and discussion

We conclude that the elimination distance to a forest characterizes the Feedback Vertex Set problem in terms of polynomial kernelization. For a minor-closed graph family $\mathcal{G}$, the problem admits a polynomial kernel in the size of a $\mathcal{G}$-modulator if and only if $\mathcal{G}$ has bounded elimination distance to a forest, assuming NP $\not\subseteq$ coNP/poly. In particular, this implies that Feedback Vertex Set does not admit a polynomial kernel in the deletion distance to a graph of constant bridge-depth under the mentioned hardness assumption, since graphs of constant bridge-depth can have arbitrarily large elimination distance to a forest. For example, the graph obtained from a path on n vertices by attaching a new triangle onto each vertex has constant bridge-depth, but its elimination distance to a forest is $\Omega(\log n)$.

We generalized our lower bound to other $\mathcal{F}$-Minor Free Deletion problems where $\mathcal{F}$ contains only biconnected planar graphs on at least three vertices. It remains unknown whether such a lower bound also generalizes to collections of graphs $\mathcal{F}$ that contain non-planar graphs.

An interesting open problem is whether similar polynomial kernels can be obtained for other $\mathcal{F}$-Minor Free Deletion problems. Regarding the field of fixed-parameter tractable algorithms, it was recently shown [31] that for any set $\mathcal{F}$ of connected graphs, $\mathcal{F}$-Minor Free Deletion admits an FPT algorithm when parameterized by the elimination distance to an $\mathcal{F}$-minor free graph (or even $\mathcal{H}$-treewidth when $\mathcal{H}$ is the class of $\mathcal{F}$-minor free graphs). This generalizes known FPT algorithms for the natural parameterization by solution size. Regarding polynomial kernels, $\mathcal{F}$-Minor Free Deletion problems admit a polynomial kernel in the solution size when $\mathcal{F}$ contains a planar graph [24]. Do polynomial kernels exist when the problem is parameterized by a modulator to a graph of constant elimination distance to being $\mathcal{F}$-minor free?

For completeness we remark that, when using the elimination distance to an $\mathcal{F}$-minor free graph as the parameter (rather than the size of a modulator whose removal makes this elimination distance constant), it follows from known results that there is no polynomial kernel for $\mathcal{F}$-Minor Free Deletion unless NP $\subseteq$ coNP/poly. This holds already for the concrete case of Vertex Cover parameterized by treedepth, which follows from the lower bound given by Bodlaender, Downey, Fellows, and Hermelin [6, Lemma 5]. They give a lower bound for Independent Set parameterized by treewidth, which is equivalent to Vertex Cover parameterized by treewidth, and the same proof goes through when replacing treewidth by treedepth; see [21, §4.2].

Data availability

No data was used for the research described in the article.

Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 803421, ReduceSearch).

References

[1] F.N. Abu-Khzam, M.R. Fellows, M.A. Langston, W.H. Suters, Crown structures for vertex cover kernelization, Theory Comput. Syst. 41 (3) (2007) 411–430, http://dx.doi.org/10.1007/s00224-007-1328-0.

[2] T. Biedl, G. Kant, A better heuristic for orthogonal graph drawings, Comput. Geom. 9 (3) (1998) 159–180, http://dx.doi.org/10.1016/S0925-7721(97)00026-6.

[3] H.L. Bodlaender, A linear-time algorithm for finding tree-decompositions of small treewidth, SIAM J. Comput. 25 (6) (1996) 1305–1317, http://dx.doi.org/10.1137/S0097539793251219.

[4] H.L. Bodlaender, Kernelization: New upper and lower bound techniques, in: J. Chen, F.V. Fomin (Eds.), Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10–11, 2009, Revised Selected Papers, in: Lecture Notes in Computer Science, vol. 5917, Springer, 2009, pp. 17–37, http://dx.doi.org/10.1007/978-3-642-11269-0_2.

[5] H.L. Bodlaender, M. Cygan, S. Kratsch, J. Nederlof, Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth, Inform. and Comput. 243 (2015) 86–111, http://dx.doi.org/10.1016/j.ic.2014.12.008.

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

[7] H.L. Bodlaender, T.C. van Dijk, A cubic kernel for feedback vertex set and loop cutset, Theory Comput. Syst. 46 (3) (2010) 566–597, http://dx.doi.org/10.1007/s00224-009-9234-2.

[8] M. Bougeret, B.M.P. Jansen, I. Sau, Bridge-depth characterizes which minor-closed structural parameterizations of vertex cover admit a polynomial kernel, SIAM J. Discrete Math. 36 (4) (2022) 2737–2773, http://dx.doi.org/10.1137/21m1400766.

[9] M. Bougeret, I. Sau, How much does a treedepth modulator help to obtain polynomial kernels beyond sparse graphs? Algorithmica 81 (10) (2019) 4043–4068, http://dx.doi.org/10.1007/s00453-018-0468-8.

[10] B. Bui-Xuan, J.A. Telle, M. Vatshelle, Feedback vertex set on graphs of low cliquewidth, in: J. Fiala, J. Kratochvíl, M. Miller (Eds.), Combinatorial Algorithms, 20th International Workshop, IWOCA 2009, Hradec Nad Moravicí, Czech Republic, June 28–July 2 2009, Revised Selected Papers, in: Lecture Notes in Computer Science, vol. 5874, Springer, 2009, pp. 113–124, http://dx.doi.org/10.1007/978-3-642-10217-2_14.

[11] J. Bulian, A. Dawar, Fixed-parameter tractable distances to sparse graph classes, Algorithmica 79 (1) (2017) 139–158, http://dx.doi.org/10.1007/s00453-016-0235-7.

[12] K. Burrage, V. Estivill-Castro, M.R. Fellows, M.A. Langston, S. Mac, F.A. Rosamond, The undirected feedback vertex set problem has a poly(k) kernel, in: H.L. Bodlaender, M.A. Langston (Eds.), Parameterized and Exact Computation, Second International Workshop, IWPEC 2006, ZÜRich, Switzerland, September 13–15, 2006, Proceedings, in: Lecture Notes in Computer Science, vol. 4169, Springer, 2006, pp. 192–202, http://dx.doi.org/10.1007/11847250_18.

[13] J. Chen, I.A. Kanj, W. Jia, Vertex cover: Further observations and further improvements, J. Algorithms 41 (2) (2001) 280–301, http://dx.doi.org/10.1006/jagm.2001.1186.

[14] M. Chlebík, J. Chlebíková, Crown reductions for the minimum weighted vertex cover problem, Discrete Appl. Math. 156 (3) (2008) 292–312, http://dx.doi.org/10.1016/j.dam.2007.03.026.

[15] B. Chor, M. Fellows, D.W. Juedes, Linear kernels in linear time, or how to save k colors in $O(n^2)$ steps, in: J. Hromkovic, M. Nagl, B. Westfechtel (Eds.), Graph-Theoretic Concepts in Computer Science, 30th International Workshop, WG 2004, Bad Honnef, Germany, June 21–23, 2004, Revised Papers, in: Lecture Notes in Computer Science, vol. 3353, Springer, 2004, pp. 257–269, http://dx.doi.org/10.1007/978-3-540-30559-0_22.

[16] J. Chuzhoy, Z. Tan, Towards tight(er) bounds for the excluded grid theorem, J. Comb. Theory Ser. B 146 (2021) 219–265, http://dx.doi.org/10.1016/j.jctb.2020.09.010.

[17] M. Cygan, F.V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, Springer, 2015, http://dx.doi.org/10.1007/978-3-319-21275-3.

[18] M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J.M.M. van Rooij, J.O. Wojtaszczyk, Solving connectivity problems parameterized by treewidth in single exponential time, ACM Trans. Algorithms 18 (2) (2022) 17:1–17:31, http://dx.doi.org/10.1145/3506707.

[19] H. Dell, D. van Melkebeek, Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses, J. ACM 61 (4) (2014) http://dx.doi.org/10.1145/2629620.

[20] H. Donkers, B.M.P. Jansen, Preprocessing to reduce the search space: Antler structures for feedback vertex set, in: L. Kowalik, M. Pilipczuk, P. Rzazewski (Eds.), Graph-Theoretic Concepts in Computer Science - 47th International Workshop, WG 2021, Warsaw, Poland, June 23–25, 2021, Revised Selected Papers, in: Lecture Notes in Computer Science, vol. 12911, Springer, 2021, pp. 1–14, http://dx.doi.org/10.1007/978-3-030-86838-3_1.

[21] M.R. Fellows, B.M.P. Jansen, F.A. Rosamond, Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity, European J. Combin. 34 (3) (2013) 541–566, http://dx.doi.org/10.1016/j.ejc.2012.04.008.

[22] M.R. Fellows, D. Lokshtanov, N. Misra, M. Mnich, F.A. Rosamond, S. Saurabh, The complexity ecology of parameters: An illustration using bounded max leaf number, Theory Comput. Syst. 45 (4) (2009) 822–848, http://dx.doi.org/10.1007/s00224-009-9167-9.

[23] F.V. Fomin, D. Lokshtanov, N. Misra, G. Philip, S. Saurabh, Hitting forbidden minors: Approximation and kernelization, SIAM J. Discrete Math. 30 (1) (2016) 383–410, http://dx.doi.org/10.1137/140997889.

[24] F.V. Fomin, D. Lokshtanov, N. Misra, S. Saurabh, Planar $\mathcal{F}$-Deletion: Approximation, kernelization and optimal FPT algorithms, in: 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20–23, 2012, IEEE Computer Society, 2012, pp. 470–479, http://dx.doi.org/10.1109/FOCS.2012.62.

[25] A. Gupta, E. Lee, J. Li, P. Manurangsi, M. Wlodarczyk, Losing treewidth by separating subsets, in: T.M. Chan (Ed.), Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6–9, 2019, SIAM, 2019, pp. 1731–1749, http://dx.doi.org/10.1137/1.9781611975482.104.

[26] E.C. Hols, S. Kratsch, Smaller parameters for vertex cover kernelization, in: N. Nishimura D. Lokshtanov (Ed.), 12th International Symposium on Parameterized and Exact Computation, IPEC 2017, September 6–8, 2017, Vienna, Austria, in: LIPIcs, vol. 89, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, pp. 20:1–20:12, http://dx.doi.org/10.4230/LIPIcs.IPEC.2017.20.

[27] Y. Iwata, Linear-time kernelization for feedback vertex set, in: I. Chatzigiannakis, P. Indyk, F. Kuhn, A. Muscholl (Eds.), 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10–14, 2017, Warsaw, Poland, in: LIPIcs, vol. 80, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, pp. 68:1–68:14, http://dx.doi.org/10.4230/LIPIcs.ICALP.2017.68.

[28] Y. Iwata, Y. Kobayashi, Improved analysis of highest-degree branching for feedback vertex set, Algorithmica 83 (8) (2021) 2503–2520, http://dx.doi.org/10.1007/s00453-021-00815-w.

[29] A. Jacob, F. Panolan, V. Raman, V. Sahlot, Structural parameterizations with modulator oblivion, in: Y. Cao, M. Pilipczuk (Eds.), 15th International Symposium on Parameterized and Exact Computation, IPEC 2020, December 14–18, 2020, Hong Kong, China Virtual Conference, in: LIPIcs, vol. 180, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, pp. 19:1–19:18, http://dx.doi.org/10.4230/LIPIcs.IPEC.2020.19.

[30] B.M.P. Jansen, H.L. Bodlaender, Vertex cover kernelization revisited, Theory Comput. Syst. 53 (2) (2012) 263–299, http://dx.doi.org/10.1007/s00224-012-9393-4.

[31] B.M.P. Jansen, J.J.H. de Kroon, M. Włodarczyk, Vertex deletion parameterized by elimination distance and even less, in: V. V. Williams S. Khuller (Ed.), STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21–25, 2021, ACM, 2021, pp. 1757–1769, http://dx.doi.org/10.1145/3406325.3451068.
[32] B.M.P. Jansen, A. Pieterse, Polynomial kernels for hitting forbidden minors under structural parameterizations, Theoret. Comput. Sci. 841 (2020) 124–166, http://dx.doi.org/10.1016/j.tcs.2020.07.009.
[33] B. Jansen, V. Raman, M. Vatshelle, Parameter ecology for feedback vertex set, Tsinghua Sci. Technol. 19 (4) (2014) 387–409, http://dx.doi.org/10.1109/TST.2014.6867520.
[34] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher (Eds.), Proceedings of a Symposium on the Complexity of Computer Computations, Held March 20–22, 1972, At the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA, in: The IBM Research Symposia Series, Plenum Press, New York, 1972, pp. 85–103, http://dx.doi.org/10.1007/978-1-4684-2001-2_9.
[35] T. Kociumaka, M. Pilipczuk, Faster deterministic feedback vertex set, Inform. Process. Lett. 114 (10) (2014) 556–560, http://dx.doi.org/10.1016/j.ipl.2014.05.001.
[36] C. Kuratowski, Sur le problè me des courbes gauches en topologie, Fund. Math. 15 (1) (1930) 271–283, http://dx.doi.org/10.4064/fm-15-1-271-283.
[37] J. Li, J. Nederlof, Detecting feedback vertex sets of size k in $O^*(2.7^k)$ time, in: S. Chawla (Ed.), Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5–8, 2020, SIAM, 2020, pp. 971–989, http://dx.doi.org/10.1137/1.9781611975994.58.
[38] D. Majumdar, V. Raman, Structural parameterizations of undirected feedback vertex set: FPT algorithms and kernelization, Algorithmica 80 (9) (2018) 2683–2724, http://dx.doi.org/10.1007/s00453-018-0419-4.
[39] D. Majumdar, V. Raman, S. Saurabh, Polynomial kernels for vertex cover parameterized by small degree modulators, Theory Comput. Syst. 62 (8) (2018) 1910–1951, http://dx.doi.org/10.1007/s00224-018-9858-1.
[40] G.L. Nemhauser, L.E. Trotter Jr., Vertex packings: Structural properties and algorithms, Math. Program. 8 (1) (1975) 232–248, http://dx.doi.org/10.1007/BF01580444.
[41] J. Nesetril, P.O. de Mendez, Sparsity - Graphs, Structures, and Algorithms, in: Algorithms and combinatorics, vol. 28, Springer, 2012, http://dx.doi.org/10.1007/978-3-642-27875-4.
[42] S. Thomassé, A $4k^2$ kernel for feedback vertex set, ACM Trans. Algorithms 6 (2) (2010) 32:1–32:8, http://dx.doi.org/10.1145/1721837.1721848.