# Where Treewidth and Pathwidth Diverge: Towards a Uniform Kernel for Pathwidth-$\eta$ Deletion

**Ahmed Ghazy**
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
Saarland University, Saarbrücken, Germany

**Jakob Greilhuber**
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
Saarland University, Saarbrücken, Germany

**Tim A. Hartmann**
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

**Roohani Sharma**
Discrete Mathematics Group, Institute for Basic Science, Daejeon, South Korea

## Abstract

For a constant $\eta \geq 0$, Pathwidth-$\eta$ Deletion is the problem of deciding whether, for a given graph $G$ and integer $k$, there is a set $S \subseteq V(G)$ of size at most $k$ such that the pathwidth of $G - S$ is at most $\eta$. The problems Treewidth-$\eta$ Deletion and Treedepth-$\eta$ Deletion are defined similarly for the parameters treewidth and treedepth, respectively. A landmark result of Fomin et al. [FOCS, 2012] shows that, for any constant $\eta$, all three problems admit a kernel on $O(k^{c(\eta)})$ vertices, where $c(\eta)$ is a constant depending on $\eta$.

Giannopoulou et al. [ACM TALG, 2017] show that, in some sense, this result is optimal for Treewidth-$\eta$ Deletion: for $\eta \geq 2$ and even when parameterizing by the size of a vertex cover $M$ of the input graph, there is no kernel of size $O(|M|^{\frac{\eta+1}{2}-\varepsilon})$, for any $\varepsilon > 0$. Contrasting this result, they prove that Treedepth-$\eta$ Deletion admits a uniform polynomial kernel, that is, a kernel of size $O(k^c)$ for a constant $c$ that is independent of $\eta$.

In comparison, the question whether Pathwidth-$\eta$ Deletion admits a uniform polynomial kernel has been neglected in the literature. As treewidth and pathwidth tend to behave similarly, it is natural to expect that no uniform polynomial kernel exists when parameterizing by the size of a vertex cover. Surprisingly, we show this not to be the case. More concretely, we prove the existence of a uniform polynomial kernel for Pathwidth-$\eta$ Deletion when parameterizing by

**(1)** the solution size $k$ plus the size of a set $M$ such that $G - M$ has bounded treedepth,
**(2)** the (vertex-deletion) distance to pathwidth-1 graphs,
**(3)** the distance to the class of graphs with treedepth at most $\eta + 1$.

This pinpoints a striking difference between Pathwidth-$\eta$ Deletion and Treewidth-$\eta$ Deletion and leads us to conjecture that Pathwidth-$\eta$ Deletion admits a uniform polynomial kernel when parameterizing by the solution size $k$.

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

**Keywords and phrases** Uniform kernelization, pathwidth deletion, treedepth, elimination distance

**Digital Object Identifier** 10.4230/LIPIcs.ESA.2026.98

**Related Version** *Full Version*: `https://arxiv.org/abs/2608.09800`

**Funding** *Roohani Sharma*: Supported by the Young Scientist Fellowship of the Institute for Basic Science (IBS-R029-Y8).

**Acknowledgements** A significant part of this work was done while A. Ghazy, J. Greilhuber and T. A. Hartmann were visiting R. Sharma at the Institute for Basic Science, Daejeon.


34th Annual European Symposium on Algorithms (ESA 2026).
Editors: Philip Bille, Seth Pettie, and Sabine Storandt; Article No. 98; pp. 98:1–98:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

# 1 Introduction

The problem $\mathcal{F}$-Minor-free Deletion is defined for each finite family $\mathcal{F}$ of undirected graphs. The input is an undirected graph $G$ and a non-negative integer $k$; the question is whether there exists a set of at most $k$ vertices of $G$ whose deletion results in an $\mathcal{F}$-minor free graph, that is, a graph that has no graph from $\mathcal{F}$ as a minor. A main objective in the field of parameterized complexity is to design *fixed-parameter tractable* (FPT) algorithms, which are algorithms with running times of the form $f(p) \cdot n^{O(1)}$ where $p$ is a parameter given in the input and $n$ the input size. One of the central goals in the field is finding the smallest parameter for which an FPT algorithm exists for the considered problem. The canonical starting point is to consider parameterizing the problem by the solution size $k$. Given the expressive power of the $\mathcal{F}$-Minor-free Deletion problem, determining the parameterized complexity for every choice of $\mathcal{F}$ is nothing short of a daunting task.

Indeed, the $\mathcal{F}$-Minor-free Deletion framework encompasses any vertex-deletion problem to a minor-closed graph class.[1] This includes classical NP-hard problems like Vertex Cover (or equivalently vertex-deletion to treewidth 0), Feedback Vertex Set (or equivalently vertex-deletion to treewidth at most 1), Treewidth-$\eta$ Deletion, Pathwidth-$\eta$ Deletion, Treedepth-$\eta$ Deletion for any $\eta \geq 0$, Outerplanar Deletion (vertex-deletion to outer-planarity), Planar Vertex Deletion (vertex-deletion to planarity) and many more.

Parameterized by the solution size $k$, a non-uniform[2] FPT algorithm for $\mathcal{F}$-Minor-free Deletion can be easily obtained via any minor-testing algorithm [38, 47] because of the well-quasi-ordering of the graph class of positive instances. A uniform FPT algorithm was later given by Fomin, Lokshtanov, Misra and Saurabh [28] for the case when $\mathcal{F}$ contains a planar graph. This was later generalized by Sau, Stamoulis and Thilikos [49] to every family $\mathcal{F}$.

### Uniform Kernelization

Another main objective in the field of parameterized complexity is to design preprocessing algorithms (*kernels*) that compress a given instance into an equivalent instance of size bounded by some function (ideally polynomial) of the parameter $p$. Again, it is desirable to find the smallest choice of parameter for which the problem admits a polynomial kernel. For the solution size parameterization, in contrast to the FPT landscape, it is still a major open problem whether $\mathcal{F}$-Minor-free Deletion admits a polynomial kernel for every $\mathcal{F}$ [11, 28, 30, 36].

On the upside, assuming that $\mathcal{F}$ contains a planar graph, a seminal work by Fomin, Lokshtanov, Misra and Saurabh [28] gives a polynomial kernel for $\mathcal{F}$-Minor-free Deletion. More concretely, they show that every $\mathcal{F}$-Minor-free Deletion problem admits a kernel of size $O(k^{f(\mathcal{F})})$ for some function $f$. Note that the kernel size is not explicit in the sense that not all involved constants are known. Interestingly, the kernel size has a dependency on $\mathcal{F}$ in the exponent. This raises the question of which $\mathcal{F}$-Minor-free Deletion problems admit a *uniform* polynomial kernel; that is, a kernel of size $g(\mathcal{F}) \cdot k^c \in O(k^c)$ for some constant $c$ independent of $\mathcal{F}$.

[1] This is due to the seminal result of Robertson and Seymour [48] who showed that minor-closed graph classes are well-quasi-ordered and hence admit a finite minor characterization [4, Corollary 2.1.1].
[2] in the sense that the algorithm might be different for every $k$

In this setting, due to the celebrated Grid Minor Theorem [15, 46], the graph obtained after removing a solution has treewidth bounded by a function that depends on the size of the excluded planar graph, making the family of Treewidth-$\eta$ Deletion problems a prototypical subclass of $\mathcal{F}$-Minor-free Deletion when $\mathcal{F}$ contains a planar graph.

For any $\eta \geq 0$, in the Treewidth-$\eta$ Deletion problem, given an undirected graph $G$ and a non-negative integer $k$, the goal is to determine if there exists a set of at most $k$ vertices whose deletion results in a graph of treewidth at most $\eta$. Treedepth-$\eta$ Deletion and Pathwidth-$\eta$ Deletion are the same as Treewidth-$\eta$ Deletion where treewidth is replaced with treedepth and pathwidth, respectively. The aforementioned work of Fomin et al. [28] provides a (non-uniform) polynomial kernel, of size $g(\eta) \cdot k^{O(\eta)}$ for some function $g$, for Treewidth-$\eta$ Deletion.

Does Treewidth-$\eta$ Deletion admit a uniform polynomial kernel? For small $\eta$, that misleadingly appears to be the case. For $\eta = 0$ (which is Vertex Cover) as well as for $\eta = 1$ (which is Feedback Vertex Set), there is a kernel with $O(k^2)$ edges, which is also essentially tight, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [14, 17, 23, 52]. However, this does not extend to larger values of $\eta$. Indeed, Giannopoulou et al. [30], provide the following lower bound that rules out a uniform kernel even when parameterizing by the size of a minimum vertex cover $\mathtt{vc}$ of the input graph.

▶ **Theorem 1.1** (Giannopoulou et al. [30, Theorem 1.1]). *For any function $g$ and any $\eta \geq 2$, $\varepsilon > 0$, Treewidth-$\eta$ Deletion does not admit a kernel of size $g(\eta) \cdot \mathtt{vc}^{\frac{\eta+1}{2}-\varepsilon}$, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.*

Observe that $\mathtt{vc} \geq k$ for non-trivial problem instances, and therefore this lower bound extends to $k$ as the parameter. In contrast, the same authors give a fully explicit uniform polynomial kernel for Treedepth-$\eta$ Deletion [30]. They obtain a polynomial kernel with $2^{O(\eta^2)} \cdot k^6 \in O(k^6)$ vertices for each $\eta$. This shows a striking difference between treewidth and treedepth.

Given this contrast between these two width measures, it is natural to explore whether similar problem families admit uniform polynomial kernels. As the parameter *pathwidth* lies in between treewidth and treedepth, our work addresses the following question:

Does Pathwidth-$\eta$ Deletion admit a uniform polynomial kernel?

When $\eta = 0$ the problem Pathwidth-$\eta$ Deletion is the same as Vertex Cover, which is well-known to admit kernels with a linear number of vertices, see e.g., [1, 16, 17, 25, 41, 51]. The case $\eta = 1$ has also been studied: a kernel with a quartic number of vertices was given by Philip et al. [43], which was subsequently improved to a kernel with a quadratic number of vertices [20, 53]. For each value of $\eta$, Pathwidth-$\eta$ Deletion admits a (non-uniform) polynomial kernel simply because it is an $\mathcal{F}$-Minor-free Deletion problem where $\mathcal{F}$ contains a planar graph, namely a forest [45]; hence, the result of Fomin et al. [28] applies. As is well-known, treewidth and pathwidth behave quite similarly from an algorithmic perspective, and there are only few known scenarios where they behave differently (see also the discussion by Belmonte et al. [3]). As such, it is tempting to conjecture that, just like Treewidth-$\eta$ Deletion, also Pathwidth-$\eta$ Deletion does not admit a uniform polynomial kernel, even when parameterizing by the vertex cover size. In this work, contrary to this expectation, we show that Pathwidth-$\eta$ Deletion *does* admit a uniform polynomial kernel when parameterizing by vertex cover number. In fact, we provide results that indicate that the problem might even admit a uniform polynomial kernel when parameterizing by the solution size.

### Towards a Uniform Polynomial Kernel

Let us illustrate how our work makes progress on whether Pathwidth-$\eta$ Deletion parameterized by the solution size $k$ admits a uniform kernel. That is, the following algorithmic approach might help in eventually resolving this question.

We start with an approximate solution $M \subseteq V(G)$ of size $O(k)$ which can be obtained by using known approximation algorithms for Pathwidth-$\eta$ Deletion [28, 31]. Then, $G - M$ has a pathwidth bounded in $\eta$, but there is no bound on its size. The goal is to design rules that let us reduce the following:

**(i)** the number of connected components of $G - M$, and

**(ii)** the size of an arbitrary connected component of $G - M$.

By bounding (i) and (ii) by polynomials in $k$, one immediately obtains a polynomial kernel. This is also the general approach for the kernelization of Treewidth-$\eta$ Deletion by Fomin et al. [28]. They obtain *non-uniform* polynomial bounds.

To obtain a *uniform* polynomial kernel we need to bound (i) and (ii) by *uniform* polynomials, that is by $g(\eta) \cdot k^c$ for some function $g$ and universal constant $c$ independent of $\eta$. This is something Giannopoulou et al. [30] achieve for their uniform kernel for the Treedepth-$\eta$ Deletion problem.

Regarding (i) and (ii) for Pathwidth-$\eta$ Deletion we show the following.

**(i)** We resolve (i). That is, we provide a preprocessing algorithm that reduces the number of connected components of $G - M$ to a *uniform polynomial.* As a corollary, we obtain that Pathwidth-$\eta$ Deletion, for $\eta \geq 0$, admits a kernel of size $g(\eta) \cdot \mathtt{vc}^3 \in O(\mathtt{vc}^3)$. This already shows a striking contrast between treewidth and pathwidth since, by Theorem 1.1, no such result is possible for Treewidth-$\eta$ Deletion.

**(ii)** We make substantial progress towards (ii). That is, we provide a preprocessing algorithm that, in many albeit not all cases, bounds the size of a connected component of $G - M$ by a uniform polynomial. To explain this, let $C$ be a connected component of $G - M$. We give reduction rules that can reduce the size of $C$ assuming that (a) $C$ contains a long path in which every vertex has degree 2 or (b) the treedepth of $C$ is bounded by *any function* of $\eta$. Therefore, a component that we cannot deal with has unbounded treedepth and does not contain a long path in which every vertex has degree 2.

These findings lead us to conjecture that Pathwidth-$\eta$ Deletion admits a uniform polynomial kernel.

▶ **Conjecture 1.2.** *There is a function $g$ and a constant $c$ such that for each $\eta \geq 0$,* Pathwidth-$\eta$ Deletion *has a kernel of size $g(\eta) \cdot k^c$.*

Our results already show that the kernelization complexity of Pathwidth-$\eta$ Deletion is closer to Treedepth-$\eta$ Deletion than it is to Treewidth-$\eta$ Deletion. We believe that the approaches we develop in this work are useful in settling Conjecture 1.2, positively or negatively.

- A positive resolution might follow the aforementioned path and extend our uniform polynomial bound for (ii) to all connected components of $G - M$ without needing any further restrictions.
- A negative resolution might derive a hardness result based on large connected components that are not covered by our approach for bounding (ii). In that case, it would be interesting to find the smallest parameter for which uniform kernels are possible. Our results already provide a good start.

### Our Results

Let us present our main results formally. For a graph class $\mathcal{G}$, we consider Pathwidth-$\eta$ Deletion parameterized by the size of a vertex set $M$ (called a *modulator*) such that $G - M$ is in the graph class $\mathcal{G}$. For technical reasons we assume that $M$ is part of the input, but we remark that typically this assumption is without loss of generality in the sense that for many graph classes, an approximately optimal modulator can be computed in polynomial-time.

> Pathwidth-$\eta$ Deletion/`dist`-$\mathcal{G}$
>
> Input: Graph $G$, integer $k$, a vertex set $M \subseteq V(G)$ such that $G - M$ is in $\mathcal{G}$
>
> Parameter: $|M|$
>
> Question: Is there a set $S \subseteq V(G)$ of size at most $k$ such that $G - S$ has pathwidth at most $\eta$?

Ideally, one would obtain a uniform polynomial kernel for Pathwidth-$\eta$ Deletion parameterized by the solution size $k$, i.e., for Pathwidth-$\eta$ Deletion/`dist`-$\mathcal{G}$ where $\mathcal{G} = \mathcal{G}_{\mathrm{pw}\leq\eta}$ is the set of graphs of pathwidth at most $\eta$. In this work, we obtain a uniform polynomial kernel if we additionally restrict the graph class $\mathcal{G}$ to graphs that have bounded elimination distance to graphs of pathwidth 1. Graphs with pathwidth at most one are forests in which each connected component is a caterpillar [2, Lemma 2.4].

Here, the *elimination distance* to $\mathcal{H}$ [12,13] is a generalization of the parameter treedepth. Recall that the treedepth is the minimum depth of an elimination forest whose leaves correspond to single isolated vertices. Now, roughly speaking, the *elimination distance to a graph class* $\mathcal{H}$ follows the same concept, but the leaves of the elimination forest now correspond to graphs in $\mathcal{H}$. We refer to Definition 2.2 for details. For a graph class $\mathcal{H}$ and graph $G$, we write $\mathsf{ed}_{\mathcal{H}}(G)$ to denote the elimination distance of $G$ to $\mathcal{H}$. Next, we define the graph class used in our main result.

▶ **Definition 1.3.** *For integers $i, j \geq 0$, let $\mathcal{G}_{\mathrm{tw}\leq i}$ ($\mathcal{G}_{\mathrm{pw}\leq i}$, $\mathcal{G}_{\mathrm{td}\leq i}$) be the class of graphs with treewidth (pathwidth, treedepth) at most $i$. Moreover, let $\mathcal{G}^j$ be the class of graphs $G$ with $\mathsf{ed}_{\mathcal{G}_{\mathrm{pw}\leq 1}}(G) \leq j$ and let $\mathcal{G}^j_{\mathrm{pw}\leq i} = \mathcal{G}_{\mathrm{pw}\leq i} \cap \mathcal{G}^j$.*

Now we are ready to state our main technical result.

▶ **Main Theorem 1.** *For some function $g$ and all constants $\eta, \beta \geq 0$,* Pathwidth-$\eta$ Deletion/`dist`-$\mathcal{G}^{\beta}_{\mathrm{pw}\leq\eta}$ *admits a kernel on $g(\eta, \beta) \cdot |M|^{60}$ vertices.*

*Moreover, given an instance $(G, k, M)$ of* Pathwidth-$\eta$ Deletion/`dist`-$\mathcal{G}^{\beta}_{\mathrm{pw}\leq\eta}$, *the kernelization algorithm outputs an equivalent instance $(G', k', M)$ where $G'$ is a minor of $G$, and if $(G, k, M)$ is a no-instance then $k = k'$.*

We stress that our kernel size is uniformly polynomial in $\eta$ and $\beta$. While the constant in the exponent is quite large, our goal was not to optimize it, but just to obtain an exponent that is independent of $\eta$.

It is tempting to think that Main Theorem 1 can be obtained by applying an iterative paradigm similarly to polynomial kernels for other $\mathcal{F}$-Minor-free Deletion problems, as in e.g., [11, 33, 35]. That is, to build upon (1) a reduction rule to reduce the number of connected components of $G - M$ and (2) a uniform polynomial kernel for Pathwidth-$\eta$ Deletion parameterized by the size of a modulator to $\mathcal{G}_{\mathrm{pw}\leq 1}$ only. Indeed, then one could bound the number of connected components of $G - M$ and lift (a bounded number of) roots of the elimination forest of $G - M$ into the modulator. The result is an instance of Pathwidth-$\eta$ Deletion/`dist`-$\mathcal{G}^{\beta-1}_{\mathrm{pw}\leq 1}$ where the modulator remains of bounded size. After repeating this step for at most $\beta$ rounds, this would yield an instance of Pathwidth-$\eta$ Deletion/dist-$\mathcal{G}_{\mathrm{pw}\leq 1}$. Then applying the kernel for Pathwidth-$\eta$ Deletion/dist-$\mathcal{G}_{\mathrm{pw}\leq 1}$ would result in a polynomial kernel.

However, this powerful lifting approach fails when aiming for a *uniform* polynomial kernel because each round increases the exponent of the kernel size. After $\beta$ rounds, the size bound of the resulting kernel depends on $\beta$ in the exponent. Therefore, it is highly non-trivial to obtain a uniform kernel for the parameterization we consider, even when already having the two ingredients (1) and (2) at hand (which are themselves non-trivial tasks).

## Consequences

While the graph class $\mathcal{G}^{\beta}_{\mathrm{pw}\leq\eta}$ might seem quite artificial at first glance, our Main Theorem 1 implies multiple results for more natural graph classes.

▶ **Corollary 1.4** (Of Main Theorem 1). *Each of the following problems admits a uniform polynomial kernel in $\eta, \gamma$:*

**(1)** Pathwidth-$\eta$ Deletion *parameterized by $k + |M|$ where $M$ is a modulator to $\mathcal{G}_{\mathrm{td}\leq\gamma}$ given in the input.*

**(2)** Pathwidth-$\eta$ Deletion/dist-$\mathcal{G}_{\mathrm{pw}\leq 1}$ *if $\eta \geq 1$.*

**(3)** Pathwidth-$\eta$ Deletion/dist-$\mathcal{G}_{\mathrm{td}\leq\eta+1}$.

**(4)** Pathwidth-$\eta$ Deletion/dist-$\mathcal{G}^{\eta-1}$ *if $\eta \geq 1$.*

**(5)** Pathwidth-$\eta$ Deletion/dist-$\mathcal{G}^{\mathrm{pl}\leq\gamma}_{\mathrm{pw}\leq\eta}$ *where $\mathcal{G}^{\mathrm{pl}\leq\gamma}_{\mathrm{pw}\leq\eta}$ is the class of graphs with pathwidth at most $\eta$ that do not contain a path on $\gamma + 1$ vertices as a subgraph.*

If we restrict ourselves to vertex cover parameterization, we obtain the following result with a much better kernel size as a corollary of our preprocessing rule to reduce the number of connected components.

▶ **Corollary 1.5.** *There is a function $g$ such that for every constant $\eta \geq 0$,* Pathwidth-$\eta$ Deletion *parameterized by the size of a vertex cover $M$ given in the input admits a kernel on $g(\eta) \cdot |M|^3$ vertices.*

*Moreover, given an instance $(G, k, M)$ of the problem, the kernelization algorithm outputs an instance $(G', k', M)$ where $G'$ is a subgraph of $G$, and if $(G, k, M)$ is a no-instance then $k = k'$.*

Moreover, because our kernel outputs a minor of the input graph, our result has interesting consequences which are purely graph-theoretical. These follow from a connection between kernels (that output a minor of the input graph) and the size of such obstructions, which has also been observed in other settings previously [24, 28, 30]. To explain these, we first require the notion of $k$-apices and obstructions.

▶ **Definition 1.6** (Apices and Obstructions [50]). *Let $\mathcal{G}$ be a minor-closed graph class. For each integer $k \geq 0$, define $\mathcal{G}_k$ to be the set of graphs that have a modulator to $\mathcal{G}$ of size at most $k$, clearly $\mathcal{G}_0 = \mathcal{G}$. A graph $H \in \mathcal{G}_k$ is a $k$-*apex *of $\mathcal{G}$.*

*A graph $H \notin \mathcal{G}$ is a* minor-minimal obstruction *to $\mathcal{G}$ if any proper minor of $H$ is in $\mathcal{G}$.*

Note that, for minor-closed class $\mathcal{G}$ and any $k \geq 0$, the class $\mathcal{G}_k$ is again minor-closed. A consequence of the famous Graph Minor Theorem of Robertson and Seymour [48] is that for any minor-closed graph class $\mathcal{G}$, the set of minor-minimal obstructions to $\mathcal{G}$ is finite [4, Corollary 2.1.1]. Using our kernel, we can obtain uniform bounds on the size of minor-minimal obstructions to the class of $k$-apices.

▶ **Theorem 1.7.** *There is a function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, such that for any $\eta, \beta \geq 0$ and any $k \geq 0$, any minor-minimal obstruction $H$ to the class of $k$-apices of $\mathcal{G}_{\mathrm{pw}\leq\eta}$ has at most $f(\eta, \beta) \cdot |M|^{60}$ vertices, where $M \subseteq V(H)$ is a smallest set such that $\mathsf{ed}_{\mathcal{G}_{\mathrm{pw}\leq 1}}(H - M) \leq \beta$ and $\mathrm{pw}(H - M) \leq \eta$.*

Similarly, by using our kernel of Corollary 1.5, we can bound the size of such graphs $H$ in terms of their *vertex cover number* $\mathtt{vc}(H)$.

▶ **Theorem 1.8.** *There is a function $f : \mathbb{N} \to \mathbb{N}$, such that for any $\eta \geq 0$ and any $k \geq 0$, any minor-minimal obstruction $H$ to the class of $k$-apices of $\mathcal{G}_{\mathrm{pw}\leq\eta}$ has at most $f(\eta) \cdot \mathtt{vc}(H)^3$ vertices.*

Theorem 1.8 is significant because the same statement is *not true* for treewidth, as the following result shows. Note that this result was informally stated by Giannopoulou et al. [30] as it indeed follows quite easily from a construction they use in a kernelization lower bound. For the sake of completeness, we provide a proof of the theorem in the full version.

▶ **Theorem 1.9.** *For any $\eta \geq 0$, and any $k \geq 0$, there is a minor-minimal obstruction $H$ to the graph class of $k$-apices of $\mathcal{G}_{\mathrm{tw}\leq\eta}$ with $|V(H)| = k + \eta + 1 + \binom{\mathtt{vc}(H)}{\eta+1}$ and $\mathtt{vc}\,(H) = k + \eta + 1$.*

Theorem 1.9 shows that a result analogous to Theorem 1.8 is *impossible* for treewidth: the exponent in the size of obstructions in terms of their vertex cover size must always depend on $\eta$ in a non-uniform manner.

**Related work.** To the best of our knowledge, only few cases of uniform polynomial kernels for $\mathcal{F}$-Minor-free Deletion problems parameterized by the solution size are known. Apart from the uniform kernel for Treedepth-$\eta$ Deletion [30], uniform polynomial kernels exist when $\mathcal{F}$ contains a $\theta_p$ [27] or, more generally, a $K_{2,p}$ [42]. Moreover, the problem family where for each fixed $d \geq 0$ the family $\mathcal{F}$ contains all connected graphs on $d + 1$ vertices also admits a uniform polynomial kernel [40, 54].

Another problem for which uniform kernelization has been studied is $d$-Set Packing. In this problem, given a universe $U$, a family $\mathcal{F}$ of $d$-sized subsets of $U$, and a positive integer $k$, the goal is to decide if there exist $k$ sets in the family $\mathcal{F}$ which are pairwise disjoint. This problem admits a (non-uniform) polynomial kernel of size $f(d) \cdot k^d$, for every fixed $d$ and some function $f$ [26]. A matching lower bound shows that there is no kernel for this problem which has size $g(d) \cdot k^{d-\varepsilon}$, for any $\varepsilon > 0$, and any function $g$, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [21]. For the special case $P_d$-Set Matching, where the universe is the vertex set of some underlying graph $G$ and $\mathcal{F}$ is the (only implicitly given) family of all paths of $G$ of length $d$, Dell and Marx [22] showed that a uniform kernel with respect to $d$ exists.

### Organization

We present a high-level overview of the used techniques in Section 2, and refer to the full version for further details.

## 2 Technical Overview

We now present a high-level overview of the proof of Main Theorem 1. The main tasks are to give uniform polynomial bounds for

**(i)** the number of connected components of $G - M$, and
**(ii)** the size of an arbitrary connected component of $G - M$.

## 2.1 Bounding the Number of Connected Components

As the first step, we bound the number of connected components by a uniform polynomial. We stress that this preprocessing step also works for Pathwidth-$\eta$ Deletion/$\texttt{dist-}\mathcal{G}_{\mathrm{pw}\leq\eta}$, a problem that is essentially the solution-size parameterization of Pathwidth-$\eta$ Deletion.

Kernelization enthusiasts might know similar notions to the one we will exploit, called "simplicial components" and "simplicial vertices". A simplicial component is a component of $G-M$ whose neighborhood is a clique, and a simplicial vertex is a vertex whose neighborhood is a clique. These notions have been exploited in many papers, for example [5, 7–9, 18, 19, 30], and the way we make sure we obtain components that are essentially simplicial follows the ideas of previous work on $\mathcal{F}$-Minor-free Deletion problems [19, 28, 30]. More concretely, our approach can be seen as a generalization of a reduction rule used to remove simplicial components by Bodlaender et al. [7, Rule 6] that is adjusted to work for Pathwidth-$\eta$ Deletion.

The goal of our reduction rule is as follows: we find a suitable connected component $C^\star$ of $G - M$ and remove $C^\star$ from the input graph to obtain the graph $G' = G - C^\star$ of our output instance. For the correctness, we have to make sure that removing $C^\star$ does not affect whether there is a solution $S$, that is, a set of at most $k$ vertices whose removal results in a graph of pathwidth at most $\eta$. More concretely, if there is a set $S$ of size at most $k$ such that $G' - S$ has pathwidth at most $\eta$, then we need to ensure that also $G - S$ has pathwidth at most $\eta$.

On a high level, we make sure that there exists a connected component $C$ of $G - M$ not intersected by the solution $S$ that is "similar" to $C^\star$. The basic idea is that, if a path decomposition of $G' - S$ could accommodate $C$, then it should also accommodate the similar component $C^\star$. What we mean by similarity here is that (1) $C$ should have a pathwidth at least $\mathrm{pw}(C^\star)$, and (2) given a path decomposition $\mathcal{P}$ of $G' - S$, we would like for $C$ to appear in bags that *all* contain the entire neighborhood of $C^\star$. In particular, the bags containing $C$ in $\mathcal{P}$ serve as a "blueprint" that lets us add similar bags that represent $C^\star$ without increasing the width. In that case, we obtain a path decomposition of $G - S$, as desired.

In order to enforce the neighborhood of $C^\star$ to appear in one bag, the basic idea is that a large number of disjoint paths between two vertices $x, y$ forces them to appear together in a bag, *virtually* simulating an edge $xy$. If those disjoint paths intersect sufficiently many components, then a simple property of path decompositions ensures that one of them (our "blueprint" $C$) appears only in bags that contain both $x$ and $y$.

We find the components $C^\star$ and $C$ with a simple marking scheme which we sketch now: For each $X \in \binom{M}{2} \cup \binom{M}{1}$, we consider the connected components $C'$ of $G - M$ with $X \subseteq N_G(C')$ and mark the $2|M|(\eta + 2) + 1$ such components having the largest pathwidth. If the number of connected components of $G - M$ is larger than $|M|^2 \cdot (2|M|(\eta + 2) + 1)$, at least one component is not marked; this component is the sought-after component $C^\star$.

Let us now explain why this marking scheme allows us to find a suitable component $C$. For this purpose, fix any $X = \{m_1, m_2\} \subseteq N_G(C^\star) \setminus S$.[3] As $X$ did not mark the component $C^\star$, it marked $2|M|(\eta+2)+1$ components $C'$ of $G-M$ with $X \subseteq N_G(C')$. Since $M$ is a solution, we may assume $|S| \leq |M|$, and therefore at least $2|M|(\eta + 1) + 1$ components marked by $X$ do not intersect $S$. Thus, there are many disjoint $m_1$-$m_2$ paths in $G' - S$, forcing $m_1$ and $m_2$ to appear together in a bag. From Helly's theorem for trees [34, Theorem 4.1], applied in

[3] Possibly all of $N_G(C^\star)$ is selected by $S$, but we ignore this trivial case here.

the same way as a classic proof showing each clique of a graph appears in a bag of any path decomposition of it [10], we then obtain that some bag of $\mathcal{P}$ contains all vertices of $N_G(C^\star)$. So, the component $C^\star$ is essentially a simplicial component.

Now, let $m_\ell$ be the vertex of $N_G(C^\star) \setminus S$ that is introduced last, and $m_r$ be the vertex of $N_G(C^\star) \setminus S$ that is forgotten first in $\mathcal{P}$. Then, also $X = \{m_\ell, m_r\}$ marked $2|M|(\eta+2)+1$ components $C'$ of $G - M$ with $X \subseteq N_G(C')$; of which again $2|M|(\eta+1)+1$ do not intersect $S$. For each vertex of $M$ there is some bag of $\mathcal{P}$ in which it is introduced, or forgotten. In particular, at most $2|M|$ bags forget or introduce a vertex of $M$, and these contain at most $2|M|(\eta+1)$ vertices overall. Therefore, one of the components $C$ marked by $X$ is such that each bag of $\mathcal{P}$ that contains $C$ contains all vertices of $(N_G(C) \cup N_G(C^\star)) \setminus S$. Since $\mathrm{pw}(C) \geq \mathrm{pw}(C^\star)$ due to our marking scheme, we can use (copies of) a bag $X_i$ that contains $\mathrm{pw}(C)+1$ vertices of $C$ to embed $C^\star$ into the path decomposition.

Formally, we obtain the following result where the bound on the number of connected components we obtain is stated in terms of the function $f_{2.1}(x) = x^2(2x(\eta+2)+1) \in O(x^3)$.

▶ **Theorem 2.1.** *There exists an algorithm* ReduceComponents *that runs in polynomial time, and given instance $(G,k,M)$ of* Pathwidth-$\eta$ Deletion/$\texttt{dist-}\mathcal{G}_{\mathrm{pw}\leq\eta}$*, where $G-M$ has more than $f_{2.1}(|N_G(G-M)|)$ connected components, it outputs an instance $(G',k,M)$ of* Pathwidth-$\eta$ Deletion/$\texttt{dist-}\mathcal{G}_{\mathrm{pw}\leq\eta}$ *such that*
**(1)** *$(G,k,M)$ is a yes-instance if and only if $(G',k,M)$ is a yes-instance, and*
**(2)** *$G'$ is a proper subgraph of $G$.*

In Theorem 2.1, we state the bound on the number of connected components as a function of the size of $N_G(G-M)$. Note that clearly $|N_G(G-M)| \leq |M|$, but it can also be significantly smaller, which is actually exploited in a later stage of our kernelization routine (we omit the details in this overview). Theorem 2.1 yields Corollary 1.5.

▶ **Corollary 1.5.** *There is a function $g$ such that for every constant $\eta \geq 0$,* Pathwidth-$\eta$ Deletion *parameterized by the size of a vertex cover $M$ given in the input admits a kernel on $g(\eta)\cdot|M|^3$ vertices.*

*Moreover, given an instance $(G,k,M)$ of the problem, the kernelization algorithm outputs an instance $(G',k',M)$ where $G'$ is a subgraph of $G$, and if $(G,k,M)$ is a no-instance then $k=k'$.*

## 2.2 Bounding the Size of Connected Components

We proceed to the technically more challenging part of our work: bounding the size of connected components. Here, we return to Pathwidth-$\eta$ Deletion/$\texttt{dist-}\mathcal{G}^{\beta}_{\mathrm{pw}\leq\eta}$, where components of $G-M$ not only have pathwidth at most $\eta$, but additionally have elimination distance at most $\beta$ to the class of graphs with pathwidth one.

**The high-level strategy.** Our approach combines a known high-level strategy and highly non-trivial new insights. Let us first discuss the known high-level strategy, which has been, at least partly, used in previous works such as [6, 24, 27, 28, 30, 37, 42]. In this strategy, the task of bounding the size of connected components reduces to the task of reducing the number of neighbors that vertices $m \in M$ have in “nice” connected components.

The eventual goal of this strategy is to obtain an instance $(G,k,M')$, where the modulator $M'$ has bounded size, such that each connected component of $G-M'$ is a *protrusion*. That is, each connected component of $G-M'$ is a graph with constant treewidth and a boundary of constant size to the rest of the graph. Such graphs can be immediately replaced with graphs of constant size using the technique of protrusion replacement [6, 28, 29].

Towards this goal, it is useful to work with *near-protrusions*, which are not protrusions, but essentially become protrusions after removing a solution. Concretely, we preprocess the input instance $(G, k, M)$ such that each component $C$ of $G - M$ has a neighborhood that can be decomposed into a set $X$ of size at most $2(\eta + 1)$ and a set $Y$, which may be large, but any optimal solution must delete all apart from at most $\eta + 1$ vertices of $Y$. We skip the details of this standard step and refer to the full version for the details. One useful property of such a near-protrusion $C$ is that an optimal solution selects at most $3(\eta + 1)$ vertices of $C$.

Therefore, to bound the size of the components, it remains to turn these near-protrusions into actual protrusions. Here, the known strategy again provides a recipe, so that it actually suffices to bound the total degree of the vertices in $M$. Let us explain why that is helpful. Assume that the total degree of the vertices in $M$ is, say, $g(\eta, \beta) \cdot |M|^c$, for some constant $c$ independent of $\eta, \beta$ and some function $g$. Then, let $X_1, \ldots, X_t$ be a path decomposition of width $\eta$ of $G - M$. We can create the modulator $M'$ by lifting vertices from $G - M$ as follows: for every $v \in N_G(M)$, we lift a bag that contains $v$ into the modulator. Then, each connected component of $G - M'$ has at most $2(\eta + 1)$ neighbors in $M'$, and treewidth (even pathwidth) at most $\eta$. Moreover, the size of $M'$ is still bounded by a uniform polynomial in the parameter $|M|$.

To summarize, a uniform kernel immediately follows from bounding the total degree of the vertices in $M$, and we may even assume that each component of $G - M$ is a near-protrusion.

**Bounding the degree of vertices in $M$.** As we can already bound the number of connected components of $G - M$, bounding the total degree of the vertices in $M$ can be achieved by bounding, for each vertex $m \in M$ and each component $C$ of $G - M$, the number of neighbors of $m$ in $C$.

In this part of our approach, we exploit that $C$ has elimination distance at most $\beta$ to the class of graphs of pathwidth at most 1. To state this more formally, we first define a few notions. In a rooted forest $F$ containing node $n \in V(F)$, we write $\mathrm{desc}_F(n)$ for the set of *descendants* of $n$ in $F$, including $n$ itself. Now we proceed to the definition of the elimination distance [12,13]. The definition we state is similar to the one of Bougeret et al. [11, Definition 8] for a concrete instance of this parameter.

▶ **Definition 2.2** (Elimination Distance). *Let $G$ be a graph, and $\mathcal{H}$ a graph class. An $\mathcal{H}$-*elimination forest *of $G$ is a tuple $\mathcal{F} = (F, \{Y_n\}_{n \in V(F)})$, where $F$ is a rooted forest, the sets $\{Y_n\}_{n \in V(F)}$ partition $V(G)$ and*

**(1)** *for each node $n \in V(F)$ that is not a leaf of $F$, we have $Y_n = \{n\}$,*

**(2)** *for each leaf $n \in V(F)$, we have that $G[Y_n] \in \mathcal{H}$, and*

**(3)** *for each edge $v_1 v_2 \in E(G)$, where $v_1 \in Y_{n_1}$ and $v_2 \in Y_{n_2}$, we have that $n_1$ and $n_2$ are in an ancestor-descendant relationship in $F$.*

*The depth of $\mathcal{F}$ is the depth of $F$. The $\mathcal{H}$-*elimination distance $\mathsf{ed}_{\mathcal{H}}(G)$ *of $G$ is the minimum depth of any $\mathcal{H}$-elimination forest of $G$. For a node $n \in V(F)$, we write $V_n^{\mathcal{F}} = \bigcup_{n' \in \mathrm{desc}_F(n)} Y_{n'}$. An $\mathcal{H}$-elimination forest $\mathcal{F}$ of $G$ is* nice *if for each $n \in V(F)$ we have that $G[V_n^{\mathcal{F}}]$ is a connected graph. The $\mathcal{H}$-elimination forest $\mathcal{F}$ is an $\mathcal{H}$-*elimination tree *if $F$ is a rooted tree.*

When $\mathcal{H}$ is the class of graphs on a single vertex, then a nice $\mathcal{H}$-elimination forest corresponds to a nice treedepth decomposition as introduced by Reidl et al. [44, Definition 2]. We also define the related notion of $\mathcal{F}$-neighbors.

▶ **Definition 2.3** (Neighbors in an Elimination Forest). *Let $G'$ be a subgraph of a graph $G$, $\mathcal{H}$ a graph class, and $\mathcal{F} = (F, \{Y_n\}_{n \in V(F)})$ an $\mathcal{H}$-elimination forest of $G'$. A vertex $v \in V(G) \setminus V(G')$ and a node $n \in V(F)$ are* $\mathcal{F}$-neighbors *if $N_G(v) \cap Y_n \neq \emptyset$. We may simply call $v$ a neighbor of $n$ if $\mathcal{F}$ is clear from the context.*

In polynomial time, we can compute a nice $\mathcal{G}_{\mathrm{pw} \leq 1}$-elimination forest $\mathcal{F} = (F, \{Y_n\}_{n \in V(F)})$ of $C$ that has depth at most $\beta$ by observing that the class of graphs with elimination distance at most $\beta$ to the class $\mathcal{G}_{\mathrm{pw} \leq 1}$ is again a minor-closed graph class (see [13]) and by using a minor-testing algorithm [38, 47]; or by directly employing an algorithm of Bulian et al. [13, Corollary 1], see also the proof of Theorem 1 in Bougeret et al. [11]. Consider a vertex $m$ in the modulator $M$. If $m$ has many neighbors in $C$, then

**(1)** $m$ has many $\mathcal{F}$-neighbors, or

**(2)** $m$ has many neighbors in a caterpillar of $C$ that corresponds to a leaf of $\mathcal{F}$.

We obtain uniform polynomial bounds for these two cases in the following.

**(1) Bounding the number of $\mathcal{F}$-neighbors.** Let $m \in M$. We begin by elaborating on how we bound the number of $\mathcal{F}$-neighbors of $m$. Here, the idea is that we want to find some neighbor $v \in V(C)$ of $m$ such that we can safely delete the edge from $m$ to $v$. In other words, we want to find an *irrelevant edge*: an edge whose deletion does not change the answer to the instance.[4] Next, we sketch how our algorithm finds this vertex $v$, then we explain why $v$ has useful properties.

Recall that, for any $n \in V(F)$, $V_n^{\mathcal{F}}$ is the set of vertices that are introduced in the bags below and including node $n$. Moreover, for a node $n \in V(F)$, we call the set $N_G(V_n^{\mathcal{F}}) \cap V(C)$ the *ancestor-type* of $n$; observe that each node in the ancestor-type of $n$ is an ancestor of $n$ in $F$.

We begin with a marking scheme reminiscent of the one used to reduce the number of connected components. For each $X = \{m_1, m_2\} \in \binom{M}{2} \cup \binom{M}{1}$ and each depth $d$ of the elimination forest, we mark a constant (depending on $\eta, \beta$) number of nodes $n \in V(F)$ at depth $d$, such that $X \subseteq N_G(V_n^{\mathcal{F}})$. Since the depth of $F$ is bounded by the constant $\beta$, the overall number of marked nodes is uniformly bounded, and some $\mathcal{F}$-neighbors of $m$ remain unmarked.

After applying the marking scheme, a basic graph theoretic argument lets us find a parent node $p \in V(F)$ with $\alpha(\eta, \beta)$ (again a constant depending on $\eta, \beta$) number of children such that each child $c$ is not marked, but $m$ is a neighbor of $G[V_c^{\mathcal{F}}]$. By choosing $\alpha(\eta, \beta)$ large enough, we can even ensure that sufficiently many of these children have the same ancestor-type. We then choose such a child $z$ such that the pathwidth of $G[V_z^{\mathcal{F}}]$ is small (relative to the pathwidth of $G[V_c^{\mathcal{F}}]$ of the other children $c$). Let $v$ be a neighbor of $m$ in $V_z^{\mathcal{F}}$.

Our reduction rule then proceeds to delete the edge $mv$, that is, the output instance is $(G' = G - mv, k, M)$. The forward direction of safety of this rule is trivial. For the backward direction, consider some minimum solution $S$ of the output instance and some width-$\eta$ path decomposition $\mathcal{P} = X_1, \ldots, X_t$ of $G' - S$. If one of $m$, $v$ is selected, or if $m$ and $v$ appear in some bag together, then $\mathcal{P}$ is also a path decomposition of $G - S$. Therefore, we assume that this is not the case. Our goal now is showing that the path decomposition $\mathcal{P}$ can be modified without increasing the width so that $m$ and $v$ do appear in some bag together.

[4] This is in the same spirit as the notion of an *irrelevant vertex*, which was introduced by Robertson and Seymour [47]. Irrelevant vertices and irrelevant edges have since been used in more works on parameterized algorithms and kernelization, for example [28, 32, 39, 42].

Consider the connected component $H$ of the graph $G[V_z^{\mathcal{F}} \setminus S]$ that contains $v$. Our algorithm has chosen $v$ so that there are *two* graphs in $G' - S$ that are "similar" to $H$. Let us elaborate in more detail on what we mean by similar in this case.

The neighborhood of $H$ can be split into the neighborhood in $M \setminus S$ and the neighborhood in $C - S$. The first of the two graphs that is similar to $H$, call it $H_1$, is used to ensure that each vertex of $H$ meets its neighbors in $M \setminus S$, and the second of the two graphs, call it $H_2$, is used to ensure that each vertex of $H$ meets its neighbors in $C - S$. More concretely, we guarantee that

1. no vertex of $H_1$ is deleted by $S$,
2. each bag that contains a vertex of $H_1$ contains $N_G(H_1) \setminus S$,
3. each bag that contains a vertex of $H_1$ contains $N_{G-S}(H) \cap M$ (which includes $m$), and
4. $\mathrm{pw}(H_1) \geq \mathrm{pw}(H)$.

The existence of graph $H_1$ follows from our marking scheme; $H_1 = G[V_{\hat{z}}^{\mathcal{F}}]$ for a marked node $\hat{z}$.

The second graph $H_2$ is going to be $G[V_{z'}^{\mathcal{F}}]$ for a node $z'$ that is a sibling of node $z$ in the elimination forest $\mathcal{F}$. We ensure that

1. no vertex of $H_2$ is deleted by $S$,
2. each bag that contains a vertex of $H_2$ contains $N_G(H_2) \setminus S$,
3. $z'$ has the same ancestor-type as $z$,
4. $\mathrm{pw}(H_2) \geq \mathrm{pw}(H)$, and
5. $m$ is a neighbor of $H_2$.

The existence of $H_2$ follows from our choice of $z$ as one of many siblings with similar properties.

Having these graphs $H_1$, $H_2$ at hand, we eventually find a bag $X_i$ that we can copy to embed (a part of) $H$. Since the details are rather involved, we do not present them here. Overall, we obtain Lemma 2.4. We strengthen the result by actually allowing $\mathcal{F}$ to be an $\mathcal{H}$-elimination tree of $C$ for any graph class $\mathcal{H}$. The bound on the number of $\mathcal{F}$-neighbors is expressed in terms of a function $\mathsf{B}_{\mathcal{F}}(x) \in O(x^2)$, where the $O$-notation hides constant factors depending on the constants $\eta, \beta$.

▶ **Lemma 2.4.** *Let $(G, k, M)$ be an instance of* Pathwidth-$\eta$ Deletion/`dist-`$\mathcal{G}_{\mathrm{pw}\leq\eta}$*, and $C \subseteq G - M$ be a near-protrusion of $(G, k)$. Let $\mathcal{F}$ be a nice $\mathcal{H}$-elimination tree of $C$ of depth at most $\beta$ for some arbitrary graph class $\mathcal{H}$, and assume that some vertex $m \in N_G(C)$ has more than $\mathsf{B}_{\mathcal{F}}(|N_G(C)|)$ $\mathcal{F}$-neighbors.*

*There exists an algorithm* ReduceDegreeTree *that runs in polynomial time, and given $(G, k, M), C, \mathcal{F}$ as input, the algorithm outputs an instance $(G', k, M)$ of* Pathwidth-$\eta$ Deletion/`dist-`$\mathcal{G}_{\mathrm{pw}\leq\eta}$ *equivalent to $(G, k, M)$ where $G'$ is a proper subgraph of $G$.*

**(2) Bounding the number of neighbors in caterpillars.** Let $m \in M$, and let $C$ be a caterpillar graph corresponding to a leaf in the elimination forest $\mathcal{F}$. Our task is to bound the number of neighbors that $m$ has in $C$. We provide the following much stronger result that can immediately bound the size of caterpillars in terms of their neighborhood size. Here, we again express the obtained bounds in terms of a function $f_{2.5}(x) \in O(x^8)$, where the $O$-notation hides constant factors depending on the constant $\eta$.

▶ **Lemma 2.5.** *Let $(G, k, M)$ be an instance of* Pathwidth-$\eta$ Deletion/`dist-`$\mathcal{G}_{\mathrm{pw}\leq\eta}$*, $C$ be an induced subgraph of $G - M$ that is a caterpillar satisfying $|V(C)| > f_{2.5}(|N_G(C)|)$.*

*There is an algorithm* ReduceCaterpillar *that runs in polynomial time and, given $(G, k, M)$, $C$ as input, it outputs an equivalent instance $(G', k, M)$ where $G'$ is a proper minor of $G$.*

Therefore, for each $m \in M$, we bound the number of neighbors that $m$ has in caterpillars corresponding to leaves of $\mathcal{F}$, as well as the number of $\mathcal{F}$-neighbors of $m$. As a result, we bound the number of neighbors each $m \in M$ has in $G - M$, which bounds the total degree of $M$.

This concludes the construction of our kernel since now we have all the properties to apply the high-level strategy mentioned earlier. In our actual kernel, we do not rely on the big hammer of protrusion replacers in the following sense: we create our own explicit approach to reduce the size of protrusions. This allows us to obtain a uniform polynomial kernel with exact knowledge of the constants involved in the kernel size.

## References


1 Faisal N. Abu-Khzam, Michael R. Fellows, Michael A. Langston, and W. Henry Suters. Crown structures for vertex cover kernelization. *Theory Comput. Syst.*, 41(3):411–430, 2007. doi:10.1007/S00224-007-1328-0.

2 Stefan Arnborg, Andrzej Proskurowski, and Detlef Seese. Monadic second order logic, tree automata and forbidden minors. In Egon Börger, Hans Kleine Büning, Michael M. Richter, and Wolfgang Schönfeld, editors, *Computer Science Logic, 4th Workshop, CSL '90, Heidelberg, Germany, October 1-5, 1990, Proceedings*, volume 533 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 1990. doi:10.1007/3-540-54487-9_49.

3 Rémy Belmonte, Eun Jung Kim, Michael Lampis, Valia Mitsou, and Yota Otachi. Grundy distinguishes treewidth from pathwidth. *SIAM J. Discret. Math.*, 36(3):1761–1787, 2022. doi:10.1137/20M1385779.

4 Daniel Bienstock and Michael A. Langston. Algorithmic implications of the graph minor theorem. In *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*, pages 481–502. Elsevier, 1995. doi:10.1016/S0927-0507(05)80125-2.

5 Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *Siam Journal On Computing*, 25(6):1305–1317, 1996. doi:10.1137/S0097539793251219.

6 Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (meta) kernelization. *J. ACM*, 63(5):44:1–44:69, 2016. doi:10.1145/2973749.

7 Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernel bounds for structural parameterizations of pathwidth. In Fedor V. Fomin and Petteri Kaski, editors, *Algorithm Theory - SWAT 2012 - 13th Scandinavian Symposium and Workshops, Helsinki, Finland, July 4-6, 2012. Proceedings*, volume 7357 of *Lecture Notes in Computer Science*, pages 352–363. Springer, 2012. doi:10.1007/978-3-642-31155-0_31.

8 Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Preprocessing for treewidth: A combinatorial analysis through kernelization. *SIAM J. Discret. Math.*, 27(4):2108–2142, 2013. doi:10.1137/120903518.

9 Hans L. Bodlaender, Arie M. C. A. Koster, and Frank van den Eijkhof. Preprocessing rules for triangulation of probabilistic networks. *Comput. Intell.*, 21(3):286–305, 2005. doi:10.1111/J.1467-8640.2005.00274.X.

10 Hans L. Bodlaender and Rolf H. Möhring. The pathwidth and treewidth of cographs. *SIAM J. Discret. Math.*, 6(2):181–188, 1993. doi:10.1137/0406014.

11 Marin Bougeret, Eric Brandwein, and Ignasi Sau. Kernelization dichotomies for hitting minors under structural parameterizations. In Meena Mahajan, Florin Manea, Annabelle McIver, and Kim Thang Nguyen, editors, *43rd International Symposium on Theoretical Aspects of Computer Science, STACS 2026, Grenoble, France, March 9-13, 2026*, LIPIcs, pages 17:1–17:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026. doi:10.4230/LIPIcs.STACS.2026.17.

12 Jannis Bulian and Anuj Dawar. Graph isomorphism parameterized by elimination distance to bounded degree. *Algorithmica. An International Journal in Computer Science*, 75(2):363–382, 2016. doi:10.1007/S00453-015-0045-3.

13 Jannis Bulian and Anuj Dawar. Fixed-parameter tractable distances to sparse graph classes. *Algorithmica. An International Journal in Computer Science*, 79(1):139–158, 2017. doi:10.1007/S00453-016-0235-7.

14 Jonathan F. Buss and Judy Goldsmith. Nondeterminism within P. *SIAM J. Comput.*, 22(3):560–572, 1993. doi:10.1137/0222038.

15 Chandra Chekuri and Julia Chuzhoy. Polynomial bounds for the grid-minor theorem. *J. ACM*, 63(5):40:1–40:65, 2016. doi:10.1145/2820609.

16 Jianer Chen, Iyad A. Kanj, and Weijia Jia. Vertex cover: Further observations and further improvements. *J. Algorithms*, 41(2):280–301, 2001. doi:10.1006/JAGM.2001.1186.

17 Benny Chor, Mike Fellows, and David W. Juedes. Linear kernels in linear time, or how to save $k$ colors in $O(n^2)$ steps. In Juraj Hromkovic, Manfred Nagl, and Bernhard Westfechtel, editors, *Graph-Theoretic Concepts in Computer Science, 30th International Workshop,WG 2004, Bad Honnef, Germany, June 21-23, 2004, Revised Papers*, Lecture Notes in Computer Science, pages 257–269. Springer, 2004. doi:10.1007/978-3-540-30559-0_22.

18 François Clautiaux, Jacques Carlier, Aziz Moukrim, and Stéphane Nègre. New lower and upper bounds for graph treewidth. In Klaus Jansen, Marian Margraf, Monaldo Mastrolilli, and José D. P. Rolim, editors, *Experimental and Efficient Algorithms, Second International Workshop, WEA 2003, Ascona, Switzerland, May 26-28, 2003, Proceedings*, volume 2647 of *Lecture Notes in Computer Science*, pages 70–80. Springer, 2003. doi:10.1007/3-540-44867-5_6.

19 Marek Cygan, Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. On the hardness of losing width. *Theory of Computing Systems*, 54(1):73–82, 2014. doi:10.1007/S00224-013-9480-1.

20 Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. An improved FPT algorithm and a quadratic kernel for pathwidth one vertex deletion. *Algorithmica*, 64(1):170–188, 2012. doi:10.1007/S00453-011-9578-2.

21 Holger Dell and Dániel Marx. Kernelization of packing problems. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 68–81. SIAM, 2012. doi:10.1137/1.9781611973099.6.

22 Holger Dell and Dániel Marx. Kernelization of packing problems. *CoRR*, abs/1812.03155, 2018. arXiv:1812.03155.

23 Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM*, 61(4):23:1–23:27, 2014. doi:10.1145/2629620.

24 Huib Donkers, Bart M. P. Jansen, and Michal Wlodarczyk. Preprocessing for outerplanar vertex deletion: An elementary kernel of quartic size. *Algorithmica*, 84(11):3407–3458, 2022. doi:10.1007/S00453-022-00984-2.

25 Michael R. Fellows, Lars Jaffke, Aliz Izabella Király, Frances A. Rosamond, and Mathias Weller. What is known about vertex cover kernelization? In Hans-Joachim Böckenhauer, Dennis Komm, and Walter Unger, editors, *Adventures Between Lower Bounds and Higher Altitudes - Essays Dedicated to Juraj Hromkovič on the Occasion of His 60th Birthday*, volume 11011 of *Lecture Notes in Computer Science*, pages 330–356. Springer, 2018. doi:10.1007/978-3-319-98355-4_19.

26 Michael R. Fellows, Christian Knauer, Naomi Nishimura, Prabhakar Ragde, Frances A. Rosamond, Ulrike Stege, Dimitrios M. Thilikos, and Sue Whitesides. Faster fixed-parameter tractable algorithms for matching and packing problems. *Algorithmica*, 52(2):167–176, 2008. doi:10.1007/S00453-007-9146-Y.

27 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, Geevarghese Philip, and Saket Saurabh. Hitting forbidden minors: Approximation and kernelization. *SIAM J. Discret. Math.*, 30(1):383–410, 2016. doi:10.1137/140997889.

28 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar $F$-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*, pages 470–479. IEEE Computer Society, 2012. doi:10.1109/FOCS.2012.62.

**29** Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Bidimensionality and kernels. *SIAM J. Comput.*, 49(6):1397–1422, 2020. doi:10.1137/16M1080264.

**30** Archontia C. Giannopoulou, Bart M. P. Jansen, Daniel Lokshtanov, and Saket Saurabh. Uniform kernelization complexity of hitting forbidden minors. *ACM Trans. Algorithms*, 13(3):35:1–35:35, 2017. doi:10.1145/3029051.

**31** Anupam Gupta, Euiwoong Lee, Jason Li, Pasin Manurangsi, and Michal Wlodarczyk. Losing treewidth by separating subsets. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1731–1749. SIAM, 2019. doi:10.1137/1.9781611975482.104.

**32** Pinar Heggernes, Pim van 't Hof, Daniel Lokshtanov, and Christophe Paul. Obtaining a bipartite graph by contracting few edges. *SIAM J. Discret. Math.*, 27(4):2143–2156, 2013. doi:10.1137/130907392.

**33** Eva-Maria C. Hols, Stefan Kratsch, and Astrid Pieterse. Elimination distances, blocking sets, and kernels for vertex cover. *SIAM Journal on Discrete Mathematics*, 36(3):1955–1990, 2022. doi:10.1137/20M1335285.

**34** W. A. Horn. Three results for trees, using mathematical induction. *Journal of Research of the National Bureau of Standards, Section B: Mathematical Sciences*, 76B(1-2):39, January 1972. doi:10.6028/jres.076B.002.

**35** Bart M. P. Jansen and Astrid Pieterse. Polynomial kernels for hitting forbidden minors under structural parameterizations. *Theor. Comput. Sci.*, 841:124–166, 2020. doi:10.1016/J.TCS.2020.07.009.

**36** Bart M. P. Jansen and Michal Wlodarczyk. Lossy planarization: A constant-factor approximate kernelization for planar vertex deletion. *SIAM J. Comput.*, 54(1):1–91, 2025. doi:10.1137/22M152058X.

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

**38** Tuukka Korhonen, Michal Pilipczuk, and Giannos Stamoulis. Minor containment and disjoint paths in almost-linear time. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*, pages 53–61. IEEE, 2024. doi:10.1109/FOCS61266.2024.00014.

**39** Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. *J. ACM*, 67(3):16:1–16:50, 2020. doi:10.1145/3390887.

**40** Mithilesh Kumar and Daniel Lokshtanov. A 2lk kernel for l-component order connectivity. In Jiong Guo and Danny Hermelin, editors, *11th International Symposium on Parameterized and Exact Computation, IPEC 2016, Aarhus, Denmark, August 24-26, 2016*, volume 63 of *LIPIcs*, pages 20:1–20:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.IPEC.2016.20.

**41** Michael Lampis. A kernel of order 2 k-c log k for vertex cover. *Inf. Process. Lett.*, 111(23-24):1089–1091, 2011. doi:10.1016/J.IPL.2011.09.003.

**42** William Lochet and Roohani Sharma. Uniform polynomial kernel for deletion to $K_{2,p}$ minor-free graphs. In Julián Mestre and Anthony Wirth, editors, *35th International Symposium on Algorithms and Computation, ISAAC 2024, Sydney, Australia, December 8-11, 2024*, LIPIcs, pages 46:1–46:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ISAAC.2024.46.

**43** Geevarghese Philip, Venkatesh Raman, and Yngve Villanger. A quartic kernel for pathwidth-one vertex deletion. In Dimitrios M. Thilikos, editor, *Graph Theoretic Concepts in Computer Science - 36th International Workshop, WG 2010, Zarós, Crete, Greece, June 28-30, 2010 Revised Papers*, volume 6410 of *Lecture Notes in Computer Science*, pages 196–207, 2010. doi:10.1007/978-3-642-16926-7_19.

44 Felix Reidl, Peter Rossmanith, Fernando Sánchez Villaamil, and Somnath Sikdar. A faster parameterized algorithm for treedepth. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, volume 8572 of *Lecture Notes in Computer Science*, pages 931–942. Springer, 2014. doi:10.1007/978-3-662-43948-7_77.

45 Neil Robertson and Paul D. Seymour. Graph minors. i. excluding a forest. *J. Comb. Theory B*, 35(1):39–61, 1983. doi:10.1016/0095-8956(83)90079-5.

46 Neil Robertson and Paul D. Seymour. Graph minors. v. excluding a planar graph. *J. Comb. Theory B*, 41(1):92–114, 1986. doi:10.1016/0095-8956(86)90030-4.

47 Neil Robertson and Paul D. Seymour. Graph minors. xiii. the disjoint paths problem. *J. Comb. Theory B*, 63(1):65–110, 1995. doi:10.1006/JCTB.1995.1006.

48 Neil Robertson and Paul D. Seymour. Graph minors. xx. wagner's conjecture. *J. Comb. Theory B*, 92(2):325–357, 2004. doi:10.1016/J.JCTB.2004.08.001.

49 Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. An FPT-algorithm for recognizing $k$-apices of minor-closed graph classes. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, Saarbrücken, Germany (Virtual Conference), July 8-11, 2020*, LIPIcs, pages 95:1–95:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.95.

50 Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. $k$-apices of minor-closed graph classes. i. bounding the obstructions. *J. Comb. Theory B*, 161:180–227, 2023. doi:10.1016/J.JCTB.2023.02.012.

51 Arezou Soleimanfallah and Anders Yeo. A kernel of order 2k-c for vertex cover. *Discret. Math.*, 311(10-11):892–895, 2011. doi:10.1016/J.DISC.2011.02.014.

52 Stéphan Thomassé. A $4k^2$ kernel for feedback vertex set. *ACM Trans. Algorithms*, 6(2):32:1–32:8, 2010. doi:10.1145/1721837.1721848.

53 Dekel Tsur. Smaller kernels for two vertex deletion problems. *Inf. Process. Lett.*, 186:106493, 2024. doi:10.1016/J.IPL.2024.106493.

54 Mingyu Xiao. Linear kernels for separating a graph into components of bounded size. *J. Comput. Syst. Sci.*, 88:260–270, 2017. doi:10.1016/J.JCSS.2017.04.004.