Kernelization Using Structural Parameters on Sparse Graph Classes*

Jakub Gajarský[2], Petr Hliněný[2], Jan Obdržálek[2], Sebastian Ordyniak[2], Felix Reidl[1], Peter Rossmanith[1], Fernando Sánchez Villaamil[1], and Somnath Sikdar[1]

[1]Theoretical Computer Science, Department of Computer Science,
RWTH Aachen University, Aachen, Germany,
{reidl,rossmani,fernando.sanchez,sikdar}@cs.rwth-aachen.de.
[2]Faculty of Informatics, Masaryk University,
Brno, Czech Republic,
{gajarsky,hlineny,obdrzalek,ordyniak}@fi.muni.cz.

Abstract

Meta-theorems for polynomial (linear) kernels have been the subject of intensive research in parameterized complexity. Heretofore, there were meta-theorems for linear kernels on graphs of bounded genus, H-minor-free graphs, and H-topological-minor-free graphs. To the best of our knowledge, there are no known meta-theorems for kernels for any of the larger sparse graph classes: graphs of bounded expansion, locally bounded expansion, and nowhere dense graphs. In this paper we prove meta-theorems for these three graph classes. More specifically, we show that graph problems that have finite integer index (FII) have linear kernels on graphs of bounded expansion when parameterized by the size of a modulator to constant-treedepth graphs. For graphs of locally bounded expansion, our result yields a quadratic kernel and for nowhere dense graphs, a polynomial kernel. While our parameter may seem rather strong, we show that a linear kernel result on graphs of bounded expansion with a weaker parameter will necessarily fail to include some of the problems included in our framework. Moreover, we only require problems to have FII on graphs of constant treedepth. This allows us to prove linear kernels for problems such as Longest Path/Cycle, Exact s,t-Path, Treewidth, and Pathwidth which do not have FII in general graphs.

*Research funded by DFG-Project RO 927/12-1 “Theoretical and Practical Aspects of Kernelization”, the Czech Science Foundation under grant P202/11/0196, and the European Social Fund and the state budget of the Czech Republic under project CZ.1.07/2.3.00/30.0009 (S. Ordyniak).

1 Introduction

Data preprocessing has always been a part of algorithm design. The last decade has seen steady progress in the area of *kernelization*, an area which deals with the design of polynomial-time preprocessing algorithms. These algorithms compress an input instance of a parameterized problem into an equivalent output instance whose size is bounded by some (small) function of the parameter. Parameterized complexity theory guarantees the existence of such *kernels* for problems that are *fixed-parameter tractable*. Of special interest are cases for which the size of the output instance is bounded by a polynomial (or even linear) function of the parameter, the so-called *polynomial (or linear) kernels*.

Interest in linear kernels is not new and there have been a series of meta-theorems on linear kernels on sparse graph classes. A *meta-theorem* is a result that focuses on a problem class instead of an individual problem. In the area of graph algorithms, such meta-theorems usually have the following form: all problems that have a specific property admit an algorithm of a specific type on a specific graph class. The first steps towards such a meta-theorem appeared in a paper by Guo and Niedermeier who provided a prescription of how to design linear kernels on planar graphs for graph problems which satisfy a certain distance property [22]. Their work built on the seminal paper by Alber, Fellows, and Niedermeier who showed that Dominating Set has a linear kernel on planar graphs [1]. This was followed by the first true meta-theorem in this area by Bodlaender et al. [6] who showed that graph problems that have *finite integer index* (FII) on bounded genus graphs and satisfy a property called *quasi-compactness*, admit linear kernels on bounded genus graphs. Shortly after [6] was published, Fomin et al. [21] proved a meta-theorem for linear kernels on H-minor-free graphs, a graph class that strictly contains graphs of bounded genus. A rough statement of their main result states that any graph problem that has FII, is *bidimensional*, and satisfies a *separation property* has a linear kernel on graphs that exclude a fixed graph as minor. This result was, in turn, generalized in [25] to H-topological-minor-free graphs, which strictly contain H-minor-free graphs. Here, the problems are required to have FII and to be *treewidth-bounding*.

The keystone to all these meta-theorems is *finite integer index*. Roughly speaking, a graph problem has finite integer index if there exists a finite set $\mathcal{S}$ of graphs such that every instance of the problem can be "represented" by a member of $\mathcal{S}$. This property is the basis of the *protrusion replacement rule* whereby protrusions (pieces of the input graph satisfying certain requirements) are replaced by members of the set $\mathcal{S}$. The protrusion replacement rule is a crucial ingredient for proving small kernels. It is important to note that FII is an intrinsic property of the problem itself and is not directly related to whether it can be expressed in a certain logic. In particular, MSO$_2$ expressibility does not imply FII (see [6] for sufficiency conditions for a problem expressible in counting MSO to have FII). As an example of this phenomenon, Hamiltonian Path has FII on general graphs whereas Longest Path does not, although both are EMSO-expressible. Another point about FII is that a problem may not have FII in general graphs but may do so in restricted graph classes.

Although these meta-theorems (viewed in chronological order) steadily covered larger graph classes, the set of problems captured in their framework diminished as the other precondition(s) became stricter. For H-topological-minor-free graphs this precondition is to be treewidth bounding. A graph problem is treewidth-bounding if YES-instances have a vertex set of size linear in the parameter, the deletion of which results in a graph of bounded treewidth. Such a vertex set is called a *modulator to bounded treewidth.* Prototypical problems that satisfy this condition are FEEDBACK VERTEX SET and TREEWIDTH t-VERTEX DELETION[1], when parameterized by the solution size. A YES-instance (G, k) of FEEDBACK VERTEX SET satisfies this condition since any feedback vertex set with at most k vertices bounds the treewidth of the remaining graph by 1. Similarly, for TREEWIDTH t-VERTEX DELETION, any k sized solution bounds the treewidth by t. While the property of being treewidth-bounding appears to be a strong prerequisite to proving a meta-theorem, it is important to note that the combined properties of bidimensionality and separability (used to prove the result on H-minor-free graphs) imply that the problem is treewidth-bounding [21]. In fact, quasi-compactness may be viewed as a relaxation of treewidth-bounding. What this shows is that all meta-theorems on linear kernels for graph classes up until H-topological-minor-free graphs implicitly used a property akin to treewidth-boundedness.

Another way of viewing the meta-theorem in [25] is as follows: when parameterized by a treewidth modulator, problems that have FII have linear kernels in H-topological-minor-free graphs. A natural problem therefore is to identify the least restrictive parameter that can be used to prove a meta-theorem for linear kernels for the next well-known class in the sparse-graph hierarchy, namely, graphs of bounded expansion. This class was defined by Nešetřil and Ossona de Mendez [29] and subsumes the class of H-topological-minor-free graphs. However, a modulator to bounded treewidth does not seem to be a useful parameter for this class. Any graph class $\mathcal{G}$ can be transformed into a class $\tilde{\mathcal{G}}$ of bounded expansion by replacing every graph $G \in \mathcal{G}$ with $\tilde{G}$, obtained in turn by replacing each edge of G by a path on $|V(G)|$ vertices. This transformation changes neither the treewidth nor the feedback vertex numbers of the graphs. Hence, if a treewidth-bounding graph problem (that additionally has FII) has a linear kernel on graphs of bounded expansion then, in particular, FEEDBACK VERTEX SET and TREEWIDTH t-VERTEX DELETION have linear (vertex) kernels in general graphs. The best-known vertex kernel for FEEDBACK VERTEX SET in general graphs is quadratic [31], for TREEWIDTH t-VERTEX DELETION in general graphs is of size $k^{g(t)}$, where g is some function [20]. This strongly suggests that one would have to choose an even more restrictive parameter to prove a meta-theorem for linear kernels on graphs of bounded expansion. In particular, the parameter must not be invariant under edge subdivision. If we assume that the parameter does not increase for subgraphs, it must necessarily attain high values on paths. *Treedepth* [29] is precisely a parameter that enforces this property, since graphs of bounded treedepth are essentially degenerate graphs with no long paths. Note that bounded treedepth implies bounded treewidth.

Our contribution. We show that, assuming FII, a parameterization by the size of a modulator to bounded treedepth allows for linear kernels in linear time on graphs of bounded expansion. The same parameter yields quadratic kernels in graphs of locally bounded expansion and polynomial kernels in nowhere dense graphs, both strictly larger classes. In particular, nowhere dense graphs are the largest class that may still be called sparse [29]. In

[1] For problem definitions, see Appendix.

these results we do not require a treedepth modulator to be supplied as part of the input, as we show that it can be approximated to within a constant factor.

Furthermore, we only need FII to hold on graphs of bounded treedepth, thus including problems which do not have FII in general. Some problems that are included because of this relaxation are Longest Path/Cycle, Pathwidth and Treewidth, none of which have polynomial kernels with respect to their standard parameters, even on sparse graphs, since they admit simple AND/OR-Compositions [5]. Problems covered by our framework include Hamiltonian Path/Cycle, several variants of Dominating Set, (Connected) Vertex Cover, Chordal Vertex Deletion, Feedback Vertex Set, Induced Matching, and Odd Cycle Transversal. In particular, we cover all problems included in earlier frameworks [6, 21, 25]. We wish to emphasize, however, that this paper does not subsume these results because of our usage of a structural parameter.

To show that a parameterization by a treedepth modulator has merit outside the sparse-graph hierarchy, we extend the polynomial kernel result for Longest Path in [7] parameterized by the vertex cover number to the weaker treedepth-modulator parameter. Finally, notice that a kernelization result for Treewidth, Pathwidth or Longest Cycle on graphs of bounded expansion with a parameter closed under edge subdivision would automatically imply the same result for general graphs. This forms the crux of our belief that any relaxation of the treedepth parameter to prove a meta-theorem for linear kernels on graphs of bounded expansion will exclude problems akin to these three.

We now describe how this paper is organized. The notation that we use, the main definitions pertaining to graph classes can all be found in Section 2. Section 3 deals with the notion of finite integer index and the protrusion machinery. In Section 4, we prove our meta-theorems for graphs of bounded expansion, locally bounded expansion, and nowhere dense graphs. Section 5 deals with polynomial kernels for Longest Path with the treedepth number as parameter. We briefly discuss the parameterized ecology program and how treedepth fits into this program in Section 6. We conclude in Section 7 with some open problems. In the appendix, we define some of the graph-theoretic problems that we mention in this paper.

2 Preliminaries

We use standard graph-theoretic notation (see [13] for any undefined terminology). All our graphs are finite and simple. Given a graph G, we use $V(G)$ and $E(G)$ to denote its vertex and edge sets. For convenience we assume that $V(G)$ is a totally ordered set, and use uv instead of $\{u, v\}$ to denote the edges of G. For $X \subseteq V(G)$, we let $G[X]$ denote the subgraph of G induced by X, and we define $G - X := G[V(G) \setminus X]$. Since we will mainly be concerned with sparse graphs in this paper, we let $|G|$ denote the number of vertices in the graph G. The distance $d_G(v, w)$ of two vertices $v, w \in V(G)$ is the length (number of edges) of a shortest v, w-path in G and ∞ if v and w lie in different connected components of G. The diameter $diam(G)$ of a graph is the length of the longest shortest path between all pair of vertices in G. We denote by $\omega(G)$ the size of the largest complete subgraph of G.

The concept of neighborhood is used heavily throughout the paper. The neighborhood of a vertex $v \in V(G)$ is the set $N^G(v) = \{w \in V(G) | vw \in E(G)\}$, the degree of v is $\deg^G(v) = |N^G(v)|$, and the closed neighborhood of v is defined as $N^G[v] := N^G(v) \cup \{v\}$. We extend this naturally to sets of vertices and subgraphs: For $S \subseteq V(G)$ we denote $N^G(S)$ the set of vertices in $V(G) \setminus S$ that have at least one neighbor in S, and for a subgraph H of G we put $N^G(H) = N^G(V(H))$. Finally if X is a subset of vertices disjoint from S, then $N^G_X(S)$ is the set $N^G(S) \cap X$ (and similarly for $N^G_X(H)$). Given a graph G and a set $W \subseteq V(G)$, we also define $\partial_G(W)$ as the set of vertices in W that have a neighbor in $V \setminus W$. Note that $N^G(W) = \partial_G(V(G) \setminus W)$. A graph G is d-degenerate if every subgraph of $G' \subseteq G$ contains a vertex $v \in V(G')$ with $deg^G(v) \leqslant d$. The degeneracy of G is the smallest d such that G is d-degenerate.

In the rest of the paper we often drop the index G from all the notation if it is clear which graph is being referred to.

2.1 Minors and shallow minors

We start by defining the notion of edge contraction. Given an edge $e = uv$ of a graph G, we let G/e denote the graph obtained from G by *contracting* the edge e, which amounts to deleting the endpoints of e, introducing a new vertex w_{uv}, and making it adjacent to all vertices in $(N(u) \cup N(v)) \setminus \{u, v\}$. By contracting $e = uv$ to the vertex w, we mean that the vertex w_{uv} is renamed as w. *Subdividing* an edge is, in a sense, an opposite operation to contraction. A graph G is called a $\leqslant$*k-subdivision* of a graph H if (some) edges of H are replaced by paths of length at most $k + 1$.

A *minor* of G is a graph obtained from a subgraph of G by contracting zero or more edges. If H is a minor of G, we write $H \preceq_m G$. A graph G is H*-minor-free* if $H \not\preceq_m G$.

We next introduce the notion of a shallow minor.

Definition 1 (Shallow minor [29])**.** For an integer d, a graph H is a *shallow minor at depth d* of G if there exists a set of disjoint subsets $V_1, \dots, V_p$ of $V(G)$ such that

1. each graph $G[V_i]$ has radius at most d, meaning that there exists $v_i \in V_i$ (a *center*) such that every vertex in V_i is within distance at most d in $G[V_i]$;
2. there is a bijection $\psi \colon V(H) \to \{V_1, \dots, V_p\}$ such that for $u, v \in V(H)$, $uv \in E(H)$ iff there is an edge in G with an endpoint each in $\psi(u)$ and $\psi(v)$.

Note that if $u, v \in V(H)$, $\psi(u) = V_i$, and $\psi(v) = V_j$ then $d_G(v_i, v_j) \leqslant (2d + 1) \cdot d_H(u, v)$. The class of shallow minors of G at depth d is denoted by $G \triangledown d$. This notation is extended to graph classes $\mathcal{G}$ as well: $\mathcal{G} \triangledown d = \bigcup_{G \in \mathcal{G}} G \triangledown d$.

2.2 Parameterized problems, kernels and treewidth

In this paper we deal with parameterized problems where the value of the parameter is not explicitly specified in the input instance. This situation is slightly different from the usual case where the parameter is supplied with the input and a parameterized problem is defined as sets of tuples (x, k) as in [15]. As such, we find it convenient to adopt the definition of Flum and Grohe [18] and we feel that this is the approach one might have to choose when dealing with generalized parameters as is done in this paper.

Let Σ be a finite alphabet. A parameterization of Σ^* is a mapping $\kappa\colon \Sigma^* \to \mathbf{N}_0$ that is polynomial time computable. A parameterized problem Π is a pair (Q, κ) consisting of a set $Q \subseteq \Sigma^*$ of strings over Σ and a parameterization κ over Σ^*. A parameterized problem Π is *fixed-parameter tractable* if there exist an algorithm $\mathcal{A}$, a computable function $f\colon \mathbf{N} \to \mathbf{N}$ and a polynomial p such that for all $x \in \Sigma^*$, $\mathcal{A}$ decides x in time $f(\kappa(x)) \cdot p(|x|)$.

Definition 2 (Graph problem)**.** A *graph problem* Π is a set of pairs (G, ξ), where G is a graph and $\xi \in \mathbf{N}_0$, such that for all graphs G_1, G_2 and all $\xi \in \mathbf{N}_0$, if $G_1 \cong G_2$ then $(G_1, \xi) \in \Pi$ iff $(G_2, \xi) \in \Pi$. For a graph class $\mathcal{G}$, we define $\Pi_{\mathcal{G}}$ as the set of pairs $(G, \xi) \in \Pi$ such that $G \in \mathcal{G}$.

Definition 3 (Kernelization)**.** A *kernelization* of a parameterized problem (Q, κ) over the alphabet Σ is a polynomial-time computable function $A\colon \Sigma^* \to \Sigma^*$ such that for all $x \in \Sigma^*$, we have

1. $x \in Q$ if and only if $A(x) \in Q$,
2. $|A(x)| \leqslant g(\kappa(x))$,

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

Definition 4 (Treewidth)**.** Given a graph $G = (V, E)$, a *tree-decomposition of* G is an ordered pair $(T, \mathcal{W})$, where T is a tree and $\mathcal{W} = \{W_x \subseteq V \mid x \in V(T)\}$ is a collection of vertex sets of G, with one set for each node of the tree T such that the following hold:

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

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

In the definition above, if we restrict T to being a path, we obtain well-known notions of a *path-decomposition* and *pathwidth*. We let $\mathbf{pw}(G)$ denote the pathwidth of G. In the sequel we will often implicitly use the following fact about tree decompositions (which implies that treewidth is a parameterization in the sense of our definition if it is bounded).

Proposition 1 ([3])**.** *Given a graph G with n nodes and a constant w, it is possible to decide whether G has treewidth at most w, and if so, to compute an optimal tree decomposition of G in time $\mathcal{O}(n)$.*

2.3 Grad and graph classes of bounded expansion

Let us recall the main definitions pertaining to the notion of graphs of bounded expansion. We follow the recent book by Nešetřil and Ossona de Mendez [29].

Definition 5 (Greatest reduced average density (grad) [26, 30])**.** Let $\mathcal{G}$ be a graph class. Then the *greatest reduced average density* of $\mathcal{G}$ with *rank* d is defined as

$$\nabla_d(\mathcal{G}) = \sup_{H \in \mathcal{G} \,\triangledown\, d} \frac{|E(H)|}{|V(H)|}.$$

This notation is also used for graphs via the convention that $\nabla_d(G) := \nabla_d(\{G\})$. In particular, note that $G \,\triangledown\, 0$ denotes the set of subgraphs of G and hence $2\nabla_0(G)$ is the maximum average degree of all subgraphs of G. The *degeneracy* of G is, therefore, exactly $2\nabla_0(G)$.

Definition 6 (Bounded expansion [26])**.** A graph class $\mathcal{G}$ has *bounded expansion* if there exists a function $f\colon \mathbf{N} \to \mathbf{R}$ (called the *expansion function*) such that for all $d \in \mathbf{N}$, $\nabla_d(\mathcal{G}) \leqslant f(d)$.

If $\mathcal{G}$ is a graph class of bounded expansion with expansion function f, we say that $\mathcal{G}$ has *expansion bounded by* f. An important relation we make use of later is: $\nabla_d(G) = \nabla_0(G \,\triangledown\, d)$, i.e. the grad of G with rank d is precisely one half the maximum average degree of subgraphs of its depth d shallow minors.

Another important notion that we make use of extensively is that of treedepth. In this context, a *rooted forest* is a disjoint union of rooted trees. For a vertex x in a tree T of a rooted forest, the *height* (or *depth*) of x in the forest is the number of vertices in the path from the root of T to x. The *height of a rooted forest* is the maximum height of a vertex of the forest. The *closure* $\operatorname{clos}(\mathcal{F})$ of a rooted forest $\mathcal{F}$ is the graph with vertex set $\bigcup_{T \in \mathcal{F}} V(T)$ and edge set $\{xy\colon x \text{ is an ancestor of } y \text{ in } \mathcal{F}\}$. A *treedepth decomposition* of a graph G is a rooted forest $\mathcal{F}$ such that $G \subseteq \operatorname{clos}(\mathcal{F})$.

Definition 7 (Treedepth)**.** The *treedepth* $\mathbf{td}(G)$ of a graph G is the minimum height of any treedepth decomposition of G.

In the sequel we will often use the following fact about treedepth decompositions.

Proposition 2 ([29])**.** *Given a graph G with n nodes and a constant w, it is possible to decide whether G has treedepth at most w, and if so, to compute an optimal treedepth decomposition of G in time $\mathcal{O}(n)$.*

We list some well-known facts about graphs of bounded treedepth. Proofs that are omitted and can be found in [29].

1. If a graph has no path with more than d vertices, then its treedepth is at most d.
2. If $\mathbf{td}(G) \leqslant d$, then G has no paths with 2^d vertices and, in particular, any DFS-tree of G has depth at most $2^d - 1$.
3. If $\mathbf{td}(G) \leqslant d$, then G is d-degenerate and hence has at most $d \cdot |V(G)|$ edges.

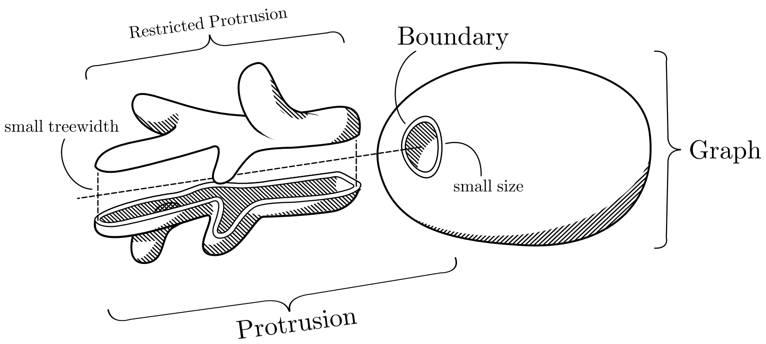

Figure 1: The anatomy of a protrusion.

4. If $\mathbf{td}(G) \leqslant d$, then $\mathbf{tw}(G) \leqslant \mathbf{pw}(G) \leqslant d-1$.

A useful way of thinking about graphs of bounded treedepth is that they are (sparse) graphs with no long paths.

For a graph G and an integer d, a *modulator to treedepth d* of G is a set of vertices $M \subseteq V(G)$ such that $\mathbf{td}(G-M) \leqslant d$. The size of a modulator is the cardinality of the set M.

Finally, we need the following well-known result on degenerate graphs.

Proposition 3 ([32])**.** *Every d-degenerate graph G with $n \geqslant d$ vertices has at most $2^d(n-d+1)$ cliques.*

3 The Protrusion Machinery

In this section, we recapitulate the main ideas of the protrusion machinery developed in [6,21].

Definition 8 (r-protrusion [6])**.** Given a graph G, a set $W \subseteq V(G)$ is a *r-protrusion* of G if $|\partial_G(W)| \leqslant r$ and $\mathbf{tw}(G[W]) \leqslant r-1$. [2] We call $\partial_G(W)$ the *boundary* and $|W|$ the *size* of the protrusion W. For an r-protrusion W, we call the set $W' = W \setminus \partial_G(W)$ the *restricted protrusion of W*.

Thus an r-protrusion in a graph is a subgraph that is separated from the rest of the graph by a small boundary and, in addition, has small treewidth. See Figure 1.

A *t-boundaried graph* is a graph G with a set $bd(G)$ of t distinguished vertices labeled 1 through t, called the *boundary*[3] or the *terminals* of G. Given a graph class $\mathcal{G}$, we let $\mathcal{G}_t$ denote the class of t-boundaried graphs from $\mathcal{G}$. If $W \subseteq V(G)$ is an r-protrusion in G, then we let G_W be the r-boundaried graph $G[W]$ with boundary $\partial_G(W)$, where the vertices of $\partial_G(W)$ are assigned labels 1 through r according to their order in B.

[2] We want the bags in a tree-decomposition of $G[W]$ to be of size at most r.

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

Definition 9 (Gluing and ungluing)**.** For t-boundaried graphs G_1 and G_2, we let $G_1 \oplus G_2$ denote the graph obtained by taking the disjoint union of G_1 and G_2 and identifying each vertex in $bd(G_1)$ with the vertex in $bd(G_2)$ with the same label. This operation is called *gluing.*

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

Note that the gluing operation entails taking the union of edges both of whose endpoints are in the boundary with the deletion of multiple edges to keep the graph simple. The ungluing operation preserves the boundary (both the vertices and the edges). For the sake of clarity, we sometimes annotate the $\oplus$ and $\ominus$ operators with the boundary as well.

Definition 10 (Replacement)**.** Let G be a graph with a t-protrusion W and let H be a t-boundaried graph. Then *replacing* W by H corresponds to the operation $(G \ominus_B G_W) \oplus_B H$.

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

Definition 11 (Finite integer index; FII)**.** Let $\Pi_{\mathcal{G}}$ be a graph problem restricted to a class $\mathcal{G}$ and let G_1, G_2 be two t-boundaried graphs in $\mathcal{G}_t$. We say that $G_1 \equiv_{\Pi_{\mathcal{G}},t} G_2$ if there exists an integer constant $\Delta_{\Pi_{\mathcal{G}},t}(G_1, G_2)$ (that depends on $\Pi_{\mathcal{G}}$, t, and the ordered pair (G_1, G_2)) such that for all t-boundaried graphs $G \in \mathcal{G}_t$ and for all $\xi \in \mathbf{N}$:

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

Note that $\Delta_{\Pi_{\mathcal{G}},t}(G_1, G_2) = -\Delta_{\Pi_{\mathcal{G}},t}(G_2, G_1)$. In the case that $(G_1 \oplus G, \xi) \notin \Pi_{\mathcal{G}}$ or $G_1 \oplus G \notin \mathcal{G}$ for *all* $G \in \mathcal{G}_t$, we set $\Delta_{\Pi_{\mathcal{G}},t}(G_1, G_2) = 0$. We say that the problem $\Pi_{\mathcal{G}}$ has *finite integer index in the class* $\mathcal{G}' \subseteq \mathcal{G}$ if, for every integer t, there are at most $g(t)$ equivalence classes of $\equiv_{\Pi_{\mathcal{G}},t}$ that contain at least one member of $\mathcal{G}'$, where g is a function that depends on t, $\Pi_{\mathcal{G}}$ and $\mathcal{G}'$.

Thus a problem $\Pi_{\mathcal{G}}$ has finite integer index in the class $\mathcal{G}' \subseteq \mathcal{G}$ iff for every integer t the equivalence relation $\equiv_{\Pi_{\mathcal{G}},t}$ *restricted to* $\mathcal{G}'$ has finite index. If a graph problem has finite integer index then its instances can be reduced by "replacing protrusions". The technique of replacing protrusions hinges on the fact that each protrusion of "large" size can be replaced by a "small" gadget from the same equivalence class as the protrusion, which consequently behaves similarly w.r.t. the problem at hand. If G_1 is replaced by a gadget G_2, then ξ changes by $\Delta_{\Pi_{\mathcal{G}},t}(G_1, G_2)$. Many problems have finite integer index in general graphs including Vertex Cover, Independent Set, Feedback Vertex Set, Dominating Set, Connected Dominating Set, Edge Dominating Set. For a more complete list see [6, 21]. Some problems that do not have finite integer index in general graphs are Connected Feedback Vertex Set, Longest Path and Longest Cycle.

Our definition above is more general than the one in [10] in that we define a problem $\Pi_{\mathcal{G}}$ to have finite integer index in a subclass $\mathcal{G}' \subseteq \mathcal{G}$ rather than in the whole class $\mathcal{G}$. The main reason behind this is the following. In Section 4, we restrict our inputs to graphs of bounded expansion but the protrusions that we replace satisfy the additional property that they have bounded treedepth. Our technique of replacing protrusions relies on a dynamic programming approach that takes one “large” protrusion of treedepth at most d and transforms it gradually by replacing small pieces from it (which are themselves protrusions) by still smaller sized representatives. We show that by systematically replacing all “large” protrusions of bounded treedepth we obtain a linear kernel. The property of finite integer index is used only for replacing protrusions which is why we require it to hold only for graphs of bounded expansion of treedepth at most some fixed constant. Our prototypical problem, Longest Path, does not have finite integer index on graph classes of bounded expansion but—as shown later—does so when we restrict the treedepth to be at most some fixed constant. Thus, this relaxed notion of FII allows us to deal with problems that do not necessarily have finite integer index on graphs of bounded expansion but do so when, in addition, the treedepth is bounded.

One must, however, be careful while replacing these pieces as we have to make sure that whatever they are replaced with must also have treedepth at most d. The following lemma will be used to show that this procedure for replacing protrusions is valid. We state Lemma 1 and Reduction Rule 1 in a more general setting than is necessary for this paper because we hope that, stated in this fashion, they might be more applicable elsewhere.

In this setting, we assume that there exists a function $\varphi\colon \mathcal{G} \to \mathbf{N}$ that maps members of a graph class to the integers. In our case, we use $\varphi \equiv \mathbf{td}$. We let $\mathcal{G}(d)$ denote the set of graphs $G \in \mathcal{G}$ for which $\varphi(G) \leqslant d$. The problems $\Pi_{\mathcal{G}}$ that we consider are such that for all $d \in \mathbf{N}$, $\Pi_{\mathcal{G}}$ has finite integer index *in* $\mathcal{G}(d)$. This means that while there can be an infinite number of equivalence classes of the relation $\equiv_{\Pi_{\mathcal{G}},t}$, for each $d \in \mathbf{N}$, at most $g(t, d)$ of these equivalence classes contain at least one graph G with $\varphi(G) \leqslant d$, where g is some function of t, d and the problem Π. For each boundary size t and $d \in \mathbf{N}$, we let $\mathcal{R}_{t,\mathcal{G}(d)}$ denote a set of graphs from $\mathcal{G}(d)$ that are representatives of these equivalence classes of $\equiv_{\Pi_{\mathcal{G}},t}$ that contain at least one graph G with $\varphi(G) \leqslant d$.

Lemma 1. *Fix $c, d, t \in \mathbf{N}$. If H is a t-boundaried graph in $\mathcal{G}(c \cdot d)$ such that $H \equiv_{\Pi_{\mathcal{G}},t} H'$ for some t-boundaried graph H' in $\mathcal{G}(d)$, then there exists $R \in \mathcal{R}_{t,\mathcal{G}(d)}$ such that $R \equiv_{\Pi_{\mathcal{G}},t} H$.*

Proof. Since $H \equiv_{\Pi_{\mathcal{G}},t} H'$, the equivalence class of $\equiv_{\Pi_{\mathcal{G}},t}$ containing H contains at least one graph from $\mathcal{G}(d)$, namely H' itself. By the definition of $R_{t,\mathcal{G}(d)}$ there exists an $R \in \mathcal{G}(d)$ that is a member of $R_{t,\mathcal{G}(d)}$ with $R \equiv_{\Pi_{\mathcal{G}},t} H$. □

For a graph problem Π that has finite integer index in the class $\mathcal{G}$, we let $\rho_{\Pi_{\mathcal{G}}}(t, d)$ denote the size of the largest representative in $\mathcal{R}_{t,\mathcal{G}(d)}$. Subscripts are omitted when the problem is clear from the context. Our reduction rule may be stated formally as follows.

Reduction Rule 1 (Protrusion replacement)**.** Let $(G, \xi) \in \Pi_{\mathcal{G}}$ and $c, d, t \in \mathbf{N}$ be constants. Suppose that $W \subseteq V(G)$ is a t-protrusion of G such that $|W| \leqslant 2\rho(t, cd)$ and suppose that $\varphi(G_W) \leqslant cd$, and $G[W] \equiv_{\Pi_{\mathcal{G}},t} H$, where $\varphi(H) \leqslant d$. Further let $R \in \mathcal{R}_{t,\mathcal{G}(d)}$ be the representative of H. The protrusion replacement rule is the following:

Reduce (G, ξ) *to* $(G', \xi') := ((G \ominus_B G_W) \oplus_B R, \xi + \Delta_{\Pi_{\mathcal{G}},t}(G_W, R))$.

The next lemma shows that this rule is indeed safe.

Proposition 4 (Safety [25])**.** *If (G', ξ') is the instance obtained from one application of the protrusion Reduction rule 1 to the instance (G, ξ) of $\Pi_{\mathcal{G}}$, then*

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

In what follows, unless otherwise stated, when applying protrusion replacement rules we will assume that for each $t \in \mathbf{N}$, we are given the set $\mathcal{R}_{t,\mathcal{G}}$ of representatives of the equivalence classes of $\equiv_{\Pi_{\mathcal{G}},t}$. Note that this makes our algorithms of Section 4 *non-uniform*. However non-uniformity is implicitly assumed in previous work that used the protrusion machinery for designing kernelization algorithms [6, 19–21], too.

4 Linear Kernels on Graphs of Bounded Expansion

In this section we show that graph-theoretic problems that have finite integer index on fixed-treedepth subclasses of graph classes of bounded expansion admit linear kernels, where the parameter is the size of a modulator to constant treedepth. Recall that a treedepth-d modulator in a graph G is a vertex set $S \subseteq V(G)$ such that $G - S$ has treedepth at most d.

Theorem 1. *Let $\mathcal{G}$ be a graph class of bounded expansion and for $p \in \mathbf{N}$, let $\mathcal{G}(p) \subseteq \mathcal{G}$ be the subclass of graphs of treedepth at most p. Let $\Pi_{\mathcal{G}}$ be a graph problem that has finite integer index on $\mathcal{G}(p)$ for each $p \in \mathbf{N}$ and let $d \in \mathbf{N}$ be a constant. Then there is an algorithm that takes as input $(G, \xi) \in \Pi_{\mathcal{G}}$ and, in time $\mathcal{O}(|G|)$, outputs an equivalent instance (G', ξ') such that $|G'| = \mathcal{O}(|S|)$, where S is an optimum treedepth-d modulator of the graph G.*

Note that we do not assume that we are given an optimal treedepth-d modulator. We show that one can approximate this to within a constant. Our proof uses an approximate modulator $S \subseteq V(G)$ to decompose $V(G)$ into vertex-disjoint sets $Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ such that

1. $S \subseteq Y_0$ and $|Y_0| = \mathcal{O}(|S|)$;
2. for $1 \leqslant i \leqslant l$, Y_i induces a collection of connected components that have exactly the same *small* neighborhood in Y_0 (to be defined later).

We then use properties of graphs of bounded expansion to show that $\ell = \mathcal{O}(|S|)$. Finally, we use the protrusion replacement rule to replace each Y_i by a graph of constant size. Every time the protrusion replacement rule is applied, ξ is modified. This results in an equivalent instance (G', ξ') such that $|G'| = \mathcal{O}(|S|)$, which is what we claim.

First let us show that one can approximate a treedepth-d modulator to within a constant.

Lemma 2. *Fix $d \in \mathbf{N}$. Given a graph G, one can in polynomial time compute a subset $S \subseteq V(G)$ such that $\mathbf{td}(G - S) \leqslant d$ and $|S|$ is at most 2^d times the size of an optimal treedepth-d modulator of G. If G is from a graph class of bounded expansion, then the same can be achieved in linear time.*

Proof. We use the fact that any DFS-tree of a graph of treedepth d has depth at most $2^d - 1$. We compute a DFS-tree of the graph G and if it has depth more than $2^d - 1$, then $\mathbf{td}(G) > d$. So, we take some path P from the root of the tree of length $2^d - 1$ and add all the 2^d vertices of P into the modulator; delete $V(P)$ from the graph and repeat. (Clearly, at least one of the vertices of P must be in any modulator.) At the end of this procedure, the DFS-tree of the remaining graph has depth at most $2^d - 1$. This gives us a tree (path) decomposition of the graph of width at most $2^d - 2$. Now use standard dynamic programming to obtain an optimum treedepth-d modulator. Since the treewidth of the remaining graph is a constant, the dynamic programming algorithm runs in time linear in the size of the graph. The overall size of the modulator has size at most 2^d times the optimal solution.

For a graph G from a class of bounded expansion, we modify the iterated depth-first search. By [26], graph classes of bounded expansion admit low treedepth coloring: Given any integer p, there exists an integer n_p such that any graph of the class can be properly vertex colored using n_p colors such that for any set of $1 \leqslant i \leqslant p$ colors, the graph induced by the vertices that receive these i colors has treedepth at most i. Such a coloring is called a p-treedepth coloring and can be computed in linear time. Here we choose $p = 2^d$ and obtain such a coloring for G using n_p colors. Let $G_1, \ldots, G_r$ denote the subgraphs induced by at most 2^d of these color classes where $r < 2^{n_p} = \mathcal{O}(1)$. Note that $\sum_j |G_j| = \mathcal{O}(|G|)$, since every vertex of G appears in at most a constant number of subgraphs. Any path in G of length $2^d - 1$ must be in some subgraph G_j, for $1 \leqslant j \leqslant r$. For each subgraph G_j, we simply construct a treedepth decomposition, find all paths of length $2^d - 1$, add their vertices into the solution and delete them from the graph. The time taken to do this for each subgraph G_j is $\mathcal{O}(|G_j|)$. The total time taken is therefore $\sum_j |G_j| = \mathcal{O}(|G|)$. □

We will make heavy use of the following lemma to prove the kernel size.

Lemma 3. *Let $G = (X, Y, E)$ be a bipartite graph. Then there are at most*

1. $2\nabla_1(G) \cdot |X|$ *vertices in Y with degree greater than* $2\nabla_1(G)$*;*
2. $(4^{\nabla_1(G)} + 2\nabla_1(G)) \cdot |X|$ *subsets $X' \subseteq X$ such that $X' = N(u)$ for some $u \in Y$.*

Proof. We construct a sequence of graphs $G_0, G_1, \ldots, G_\ell$ such that $G_i \in G \triangledown 1$ for all $0 \leqslant i \leqslant \ell$ as follows. Set $G_0 = G$, and for $0 \leqslant i \leqslant \ell - 1$ construct G_{i+1} from G_i by choosing a vertex $v \in V(G_i) \setminus X$ such that $N(v) \subseteq X$ contains two non-adjacent vertices u, w in G_i; if no such vertex v exists, stop with $\ell := i$. Set $e_{i+1} = uv$ and contract this edge to the vertex u to obtain G_{i+1}. Recall that contracting uv to u is equivalent to deleting vertex v and adding edges between each vertex in $N(v) \setminus u$ and u. It is clear from the construction that for $0 \leqslant i \leqslant \ell$, $X \subseteq V(G_i) \subseteq X \cup Y$.

This process clearly terminates, as G_{i+1} has at least one more edge between vertices of X than G_i. Note that $G_i \in G \triangledown 1$ for $0 \leqslant i \leqslant \ell$, as the edges $e_1, \ldots, e_{i-1}$ that were contracted to vertices in X in order to construct G_i had one endpoint each in X and Y, the endpoint in Y being deleted after each contraction. Thus, $e_1, \ldots, e_{i-1}$ induce a set of stars in $V(G) = V(G_0)$, and G_i is obtained from G by contracting these stars. We therefore conclude that G_i is a depth-one shallow minor of G. In particular, this implies $G_\ell[X]$ is $2\nabla_1(G)$-degenerate and has at most $2\nabla_1(G) \cdot |X|$ edges. Further, note that for each $0 \leqslant i \leqslant \ell$, $Y \cap V(G_i)$ is, by construction, still an independent set in G_i.

Let us now prove the first claim. To this end, assume that there is a vertex $v \in Y \cap V(G_\ell)$ such that $degree(v) > 2\nabla_1(G)$. We claim that $G_\ell[N(v)]$ (where $N(v) \subseteq X$) is a clique. If not, we could choose a pair of non-adjacent vertices in $G_\ell[N(v)]$ and construct a $(\ell+1)$-th graph for the sequence which would contradict the fact that G_ℓ is the last graph of the sequence. However, a clique of size $|\{v\} \cup N(v)| > 2\nabla_1(G) + 1$ is not $2\nabla_1(G)$-degenerate. Hence we conclude that no vertex of $Y \cap V(G_\ell)$ has degree larger than $2\nabla_1(G)$ in G_ℓ (and in G). Therefore the vertices of Y of degree greater than $2\nabla_1(G)$ in the graph G, if there were any, must have been deleted during the edge contractions that resulted in the graph G_ℓ. As every contraction added at least one edge between vertices in X and since $G_\ell[X]$ contains at most $2\nabla_1(G) \cdot |X|$ edges, the first claim follows.

For the second claim, consider the set $Y' = Y \cap V(G_\ell)$. The neighbourhood of every vertex $v \in Y'$ induces a clique in $G_\ell[X]$. From the degeneracy of $G_\ell[X]$, it follows that $G_\ell[X]$ has at most $2^{2\nabla_1(G)}|G_\ell[X]| = 4^{\nabla_1(G)} \cdot |X|$ cliques. Thus the number of subsets of X that are neighbourhoods of vertices in Y in G is at most $(4^{\nabla_1(G)} + 2\nabla_1(G)) \cdot |X|$, where we accounted for vertices of Y lost via contractions by the bound on the number of edges in $G_\ell[X]$. □

The following two corollaries to Lemma 3 show how it can be applied in our situation.

Corollary 1. *Let $\mathcal{G}$ be a graph-class whose expansion is bounded by a function $f\colon \mathbf{N} \to \mathbf{R}$. Suppose that for $G \in \mathcal{G}$ and $S \subseteq V(G)$, $C_1, \ldots, C_s$ are disjoint connected subgraphs of $G - S$ satisfying the following two conditions: for $1 \leqslant i \leqslant s$, $diam(G[V(C_i)]) \leqslant \delta$ and $|N_S(C_i)| > 2 \cdot f(\delta+1)$. Then $s \leqslant 2 \cdot f(\delta+1) \cdot |S|$.*

Proof. We construct an auxilliary bipartite graph $\tilde{G}$ with partite sets S and $Y = \{C_1, \ldots, C_s\}$. There is an edge between C_i and $x \in S$ iff $x \in N_S(C_i)$. Note that $\tilde{G}$ is a depth-δ shallow minor of G with branch sets $C_i, 1 \leqslant i \leqslant s$. By Lemma 3,

$$s \leqslant 2\nabla_1(\tilde{G})|S| \leqslant 2\nabla_1(G \triangledown \delta)|S| = 2\nabla_{\delta+1}(G)|S| \leqslant 2f(\delta+1)|S|. \qquad \square$$

Corollary 2. *Let $\mathcal{G}$ be a graph-class whose expansion is bounded by a function $f\colon \mathbf{N} \to \mathbf{R}$. Suppose that for $G \in \mathcal{G}$ and $S \subseteq V(G)$, $\mathcal{C}_1, \ldots, \mathcal{C}_t$ are sets of connected components of $G - S$ such that for all $C, C' \in \bigcup_i \mathcal{C}_i$ it holds that $C, C' \in \mathcal{C}_j$ for some j if and only if $N_S(C) = N_S(C')$. Let $\delta > 0$ be a bound on the diameter of the components, i.e. for all $C \in \bigcup_i \mathcal{C}_i$, $diam(G[V(C)]) \leqslant \delta$. Then there can be only at most $t \leqslant (4^{f(\delta+1)} + 2f(\delta+1)) \cdot |S|$ such sets $\mathcal{C}_i$.*

Proof. As in the proof of Corollary 1, we construct a bipartite graph $\tilde{G}$ with partite sets S and $Y = \{C_1, \ldots, C_r\}$, where the vertices C_j represent connected components in $\bigcup_i \mathcal{C}_i$ and C_j has an edge to $x \in S$ iff $x \in N_S(C_j)$. As before, $\tilde{G}$ is a shallow minor at depth δ of G

with branch sets $C_j, 1 \leqslant j \leqslant r$. By Lemma 3,

$$\begin{aligned} t \leqslant |\{S' \subseteq S \mid \exists C_i \in Y : N(C_i) = S'\}| &\leqslant (4^{\nabla_1(\tilde{G})} + 2\nabla_1(\tilde{G})) \cdot |S| \\ &\leqslant (4^{\nabla_1(G \,\triangledown\, \delta)} + 2\nabla_1(G \,\triangledown\, \delta)) \cdot |S| \\ &= (4^{\nabla_{\delta+1}(G)} + 2\nabla_{\delta+1}(G)) \cdot |S| \\ &\leqslant (4^{f(\delta+1)} + 2f(\delta+1)) \cdot |S|. \qquad \square \end{aligned}$$

Algorithm 1: Bag marking algorithm

```
Input: A graph $G$, a subset $S \subseteq V(G)$ such that $\mathbf{td}(G - S) \leqslant d$, and an integer $t > 0$.
Set $\mathcal{M} \leftarrow \emptyset$ as the set of marked bags;
for each connected component $C$ of $G - S$ such that $N_S(C) \geqslant t$ do
    Choose an arbitrary vertex $v \in V(C)$ as a root and construct a DFS-tree starting at $v$;
    Use the DFS-tree to obtain a path-decomposition $\mathcal{P}_C = (P_C, \mathcal{B}_C)$ of width at most $2^d - 2$ in which the bags are ordered from left to right;
Repeat the following loop for the path-decomposition $\mathcal{P}_C$ of every $C$;
while $\mathcal{P}_C$ contains an unprocessed bag do
    Let $B$ be the leftmost unprocessed bag of $\mathcal{P}_C$;
    Let $G_B$ denote the subgraph of $G$ induced by the vertices in the bag $B$ and in all bags to the left of it in $\mathcal{P}_C$.
    [Large-subgraph marking step]
    if $G_B$ contains a connected component $C_B$ such that $|N_S(C_B)| \geqslant t$ then
        $\mathcal{M} \leftarrow \mathcal{M} \cup \{B\}$ and remove the vertices of $B$ from every bag of $\mathcal{P}_C$;
    Bag $B$ is now processed;
return $Y_0 = S \cup V(\mathcal{M})$;
```

Lemma 4. *Let $\mathcal{G}$ be a graph class with expansion bounded by f, $G \in \mathcal{G}$ and $S \subseteq V(G)$ be a set of vertices such that $\mathbf{td}(G - S) \leqslant d$ (d a constant). There is an algorithm that runs in time $\mathcal{O}(|G|)$ and partitions $V(G)$ into sets $Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ such that the following hold:*

1. *$S \subseteq Y_0$ and $|Y_0| = \mathcal{O}(|S|)$;*
2. *for $1 \leqslant i \leqslant \ell$, Y_i induces a set of connected components of $G - Y_0$ that have the same neighborhood in Y_0 of size at most $2^{d+1} + 2 \cdot f(2^d)$;*
3. *$\ell \leqslant \left(4^{f(2^d)} + 2f(2^d)\right) \cdot |S| = \mathcal{O}(|S|)$.*

Proof. We first construct a DFS-forest $\mathcal{F}$ of $G - S$. Assume that there are q trees $T_1, \ldots, T_q$ in this forest that are rooted at $r_1, \ldots, r_q$, respectively. Since $\mathbf{td}(G - S) \leqslant d$, the height of every tree in $\mathcal{F}$ is at most $2^d - 1$. Next we construct for each T_i, where $1 \leqslant i \leqslant q$, a path decomposition of the subgraph of G induced by the vertices in T_i. Suppose that T_i has leaves $l_1, \ldots, l_s$ ordered according to their DFS-number. For $1 \leqslant j \leqslant s$, create a bag B_j containing the vertices on the unique path from l_j to r_i and string these bags together in the order $B_1, \ldots, B_s$. It is easy to verify that this is indeed a path decomposition $\mathcal{P}_i$ of $G[V(T_i)]$, that each bag has at most $2^d - 1$ vertices and that the root r_i is in every bag of the decomposition.

We now use a marking algorithm similar to the one in [25] to mark $\mathcal{O}(|S|)$ bags in the path decompositions $\mathcal{P}_1, \ldots, \mathcal{P}_q$ with the property that each marked bag can be uniquely identified with a connected subgraph of $G - S$ that has a large neighborhood in the modulator S. This algorithm is described in Figure 1 in which we set t, the size of a *large neighborhood* in S, to be $t := 2 \cdot f(2^d) + 1$. Note that there is a one-to-one correspondence between marked bags $\mathcal{M}$ and connected subgraphs with a neighborhood of size at least t in S. Moreover each connected subgraph has treedepth at most d and hence diameter at most $2^d - 1$. By Corollary 1, the number of connected subgraphs of large neighborhood and hence the number of marked bags is at most $2 \cdot f(2^d - 1 + 1) \cdot |S| = 2f(2^d) \cdot |S| = \mathcal{O}(|S|)$. We set $Y_0 := V(\mathcal{M}) \cup S$.

Now observe that each connected component in $G - Y_0$ has less than $t = 2 \cdot f(2^d) + 1$ neighbors in S. This follows because for every connected subgraph C with at least t neighbors in S, there exists a marked bag B. Importantly, the bag B was the *first* bag that was marked before the number of neighbors in S of *any* connected subgraph reached the threshold t. Hence each connected component of $G[V(C) \setminus B]$ has degree less than t in S. Since every component can be connected to at most two marked bags (in Y_0) and since each bag is of size at most $2^d - 1$, the size of the neighborhood of every component of $G - Y_0$ in Y_0 is at most $2(2^d - 1) + t \leqslant 2^{d+1} + 2 \cdot f(2^d)$.

To complete the proof, we simply cluster the connected components of $G - Y_0$ according to their neighborhoods in Y_0 to obtain the sets $Y_1, \ldots, Y_\ell$. Since each connected component of $G - S$ is of diameter $\delta \leqslant 2^d - 1$, by Corollary 2, the number ℓ of clusters is at most $\left(4^{f(2^d)} + 2f(2^d)\right) \cdot |S| = \mathcal{O}(|S|)$, as claimed. $\square$

To prove a linear kernel, all that is left to show is that each cluster Y_i, $1 \leqslant i \leqslant \ell$, can be reduced to constant size. Note that each cluster is separated from the rest of the graph via a small set of vertices in S and that each component of $G - S$ has constant treedepth. These facts enable us to use the protrusion reduction rule.

In the proof of the following lemma it will be convenient to use the following normal form of tree decompositions: A triple $(T, \{W_x \mid x \in V(T)\}, r)$ is a *nice tree decomposition* of a graph G if $(T, \{W_x \mid x \in V(T)\})$ is a tree decomposition of G, the tree T is rooted at node $r \in V(T)$, and each node of T is of one of the following four types:

1. a *leaf node*: a node having no children and containing exactly one vertex in its bag;
2. a *join node*: a node x having exactly two children y_1, y_2, and $W_x = W_{y_1} = W_{y_2}$;
3. an *introduce node*: a node x having exactly one child y, and $W_x = W_y \cup \{v\}$ for a vertex v of G with $v \notin W_y$

4. a *forget node*: a node x having exactly one child y, and $W_x = W_y \setminus \{v\}$ for a vertex v of G with $v \in W_y$.

Given a tree decomposition of a graph G of width w, one can effectively obtain in time $\mathcal{O}(|V(G)|)$ a nice tree decomposition of G with $\mathcal{O}(|V(G)|)$ nodes and of width at most w [9].

In the context of the next lemma, let $\mathcal{G}$ be a graph class of bounded expansion and, for $p \in \mathbf{N}$, let $\mathcal{G}(p)$ denote the subclass of $\mathcal{G}$ of graphs of treedepth at most p. Let $\Pi_{\mathcal{G}}$ be a graph problem that has finite integer index on $\mathcal{G}(p)$ for every fixed $p \in \mathbf{N}$. Recall that $\rho(t,d)$ denotes the size of the largest representative in $\mathcal{R}_{t,\mathcal{G}(d)}$, for the problem $\Pi_{\mathcal{G}}$.

Lemma 5. *For fixed $d, h \in \mathbf{N}$, let (G, ξ) be an instance of $\Pi_{\mathcal{G}}$ and let $S \subseteq V(G)$ be a treedepth-d modulator of G. Let $Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ be a protrusion-decomposition of G, where $S \subseteq Y_0$ and for $1 \leqslant i \leqslant \ell$, $|N_{Y_0}(Y_i)| \leqslant h$. Then one can in $\mathcal{O}(|G|)$ time obtain an equivalent instance (G', ξ') and a protrusion-decomposition $Y'_0 \uplus Y'_1 \uplus \cdots \uplus Y'_\ell$ of G' where $Y'_0 = Y_0$, and for $1 \leqslant i \leqslant \ell$ it is $|N_{Y'_0}(Y'_i)| \leqslant h$ and $|Y'_i| \leqslant \rho(d+h, d) = \mathcal{O}(1)$.*

Proof. Since $S \subseteq Y_0$ is a treedepth-d modulator, for all $1 \leqslant i \leqslant \ell$, we have $\mathbf{td}(G[Y_i]) \leqslant d$ and hence $\mathbf{tw}(G[Y_i]) \leqslant d - 1$. Moreover treedepth at most d implies diameter at most $2^d - 1$ for each component. For each index $1 \leqslant i \leqslant \ell$, our algorithm constructs a tree-decomposition of $G[Y_i \cup N(Y_i)]$ of width $d + h$ that satisfies certain properties that we mention below. The algorithm then uses this tree-decomposition to replace Y_i in a systematic manner using the protrusion replacement rule. The properties that this tree-decomposition satisfies enable the algorithm to perform this replacement in $\mathcal{O}(|Y_i \cup N(Y_i)|)$ time. The total time taken to replace all sets Y_i is $\sum_{i=1}^{\ell} |Y_i \cup N(Y_i)|$ and since by Lemma 3, $\sum_{i=1}^{\ell} |N(Y_i)| = \mathcal{O}(|Y_0|)$, the running time is indeed $\mathcal{O}(|G|)$. It therefore suffices to describe what properties our tree-decompositions satisfy and how each Y_i is replaced.

The tree-decomposition $\mathcal{T}_i = (T_i, \{W_x \mid x \in V(T_i)\})$ of width $d+h$ for $G_i := G[Y_i \cup N(Y_i)]$ satisfies the following conditions:

1. there is a node $r \in V(T_i)$ such that $N(Y_i) = W_r$;
2. the tree-decomposition is nice and the leaf bags contain one vertex.

The first condition can be achieved by simply modifying the graph G_i so that $N(Y_i)$ induces a clique, and then introducing an extra node r if no such node exists. The decomposition $\mathcal{T}_i$ is rooted at the node r. For $x \in V(T_i)$, we let G_x denote the $(d+h)$-boundaried graph induced by the vertices in the bags of the subtree of T_i rooted at x. That is,

$$G_x = G\left[\bigcup W_y\right],$$

where the union is over all $y \in V(T_i)$ that are descendants of x and $bd(G_x) = W_x$. For $x \in V(T_i)$, denote by $\Lambda(x)$ the representative of G_x in $\mathcal{R}_{d+h,\mathcal{G}(d)}$ and let $\mu(x) = \Delta_{\Pi_{\mathcal{G}},d+h}(\Lambda(x), G_x)$. Note that the treedepth of G_x is at most d and since $\Pi_{\mathcal{G}}$ has FII in $\mathcal{G}(d)$, such a representative $\Lambda(x)$ is indeed well founded. Moreover, $|\Lambda(x)| \leqslant M$ where $M := \rho(d+h, d)$ denotes the size of the largest representative in $\mathcal{R}_{d+h,\mathcal{G}(d)}$.

In order to replace Y_i, it is sufficient to know $\Lambda(r)$ and $\mu(r)$ which we will calculate in a bottom-up manner in $\mathcal{O}(|Y_i|)$ time as follows. If $y \in V(T_i)$ is a leaf node then these values can be computed in constant time. Let $x \in V(T_i)$ be a node with exactly one child y whose Λ and μ values are known. Consider the $(d+h)$-boundaried graph $G'_x := (G_x \ominus_{W_y} G_y) \oplus_{W_y} \Lambda(y)$ with $bd(G'_x) = W_x$. We claim that $G'_x \equiv_{\Pi_{\mathcal{G}},d+h} G_x$. To prove this, we need to demonstrate that for all graphs $\tilde{G}$ and all $\xi \in \mathbf{N}$,

$$(G'_x \oplus_{W_x} \tilde{G}, \xi) \in \Pi_{\mathcal{G}} \text{ if and only if } (G_x \oplus_{W_x} \tilde{G}, \xi + \mu') \in \Pi_{\mathcal{G}},$$

where $\mu' = \Delta_{\Pi_{\mathcal{G}},d+h}(G'_x, G_x)$. Now

$$\begin{aligned}(G'_x \oplus_{W_x} \tilde{G}, \xi) \in \Pi_{\mathcal{G}} &\text{ iff } ((G_x \ominus_{W_y} G_y) \oplus_{W_y} \Lambda(y)) \oplus_{W_x} \tilde{G}, \xi) \in \Pi_{\mathcal{G}} \\ &\text{ iff } ((G_x \oplus_{W_x} \tilde{G}) \ominus_{W_y} G_y) \oplus_{W_y} \Lambda(y), \xi) \in \Pi_{\mathcal{G}} \\ &\text{ iff } ((G_x \oplus_{W_x} \tilde{G}) \ominus_{W_y} G_y) \oplus_{W_y} G_y, \xi + \mu(y)) \in \Pi_{\mathcal{G}},\end{aligned}$$

where the last step follows because of $\Lambda(y) \equiv_{\Pi_{\mathcal{G}},d+h} G_y$. Since $(G_x \oplus_{W_x} \tilde{G}) \ominus_{W_y} G_y) \oplus_{W_y} G_y$ is just the graph $G_x \oplus_{W_x} \tilde{G}$, this proves our claim. In fact, $\mu' = \mu(y)$.

Observe that G'_x is of *constant* size, bounded from above by $M + |W_x| \leqslant M + d + h = \mathcal{O}(1)$. Although $\Lambda(y)$ has treedepth at most d, G'_x is not guaranteed to have treedepth at most d. In fact, G'_x can have treedepth up to $d + h$. However since $\mathbf{td}(G_x) \leqslant d$, we can use Lemma 1 to conclude that there exists $R \in \mathcal{R}_{d+h,\mathcal{G}(d)}$ with $G'_x \equiv_{\Pi_{\mathcal{G}},d+h} R$, and obtain this R in constant time since G'_x is of constant size. We set $\Lambda(x) = R$ and $\mu(x) = \mu(y) + \Delta_{\Pi,d+h}(G'_x, R)$. Note that the total time spent at node x to generate these values is a constant.

Finally consider the case when $x \in V(T_i)$ has exactly two children y_1 and y_2 whose Λ and μ values are known. Since our tree-decomposition is nice, we have $W_{y_1} = W_x = W_{y_2}$ and therefore $bd(G_{y_1}) = bd(G_{y_2}) = W_x$. Consider the $(d+h)$-boundaried graph $G''_x = \Lambda(y_1) \oplus_{W_x} \Lambda(y_2)$ with $bd(G''_x) = W_x$. Similarly as in the above case, we demonstrate that for all graphs $\tilde{G}$ and all $\xi \in \mathbf{N}$,

$$(G''_x \oplus_{W_x} \tilde{G}, \xi) \in \Pi_{\mathcal{G}} \text{ if and only if } (G_x \oplus_{W_x} \tilde{G}, \xi + \mu'') \in \Pi_{\mathcal{G}}, \quad \text{where } \mu'' = \mu(y_1) + \mu(y_2).$$

Then G''_x has size at most $2M$ which is a constant. One can therefore, again in constant time, calculate a representative $R \in \mathcal{R}_{d+h,\mathcal{G}(d)}$ of G''_x. Set $\Lambda(x) = R$ and $\mu(x) = \Delta_{\Pi,d+h}(G''_x, R)$. This shows that one can in time $\mathcal{O}(|Y_i|)$ obtain $\Lambda(r)$ and $\mu(r)$, as desired. □

With the help of this last lemma we can now prove the main theorem of this section.

Proof of Theorem 1. Given an instance (G, ξ) of Π with $G \in \mathcal{G}$ for a graph class $\mathcal{G}$ with expansion bounded by $f \colon \mathbf{N} \to \mathbf{R}$ and having fixed a constant $d \in \mathbf{N}$, we calculate a 2^d-approximation S of a minimal treedepth-d-modulator using Lemma 2. Then, we use the above Algorithm 1 to obtain the decomposition $Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ as defined in Lemma 4. Each cluster $Y_i, 1 \leqslant i \leqslant \ell$ forms a protrusion with boundary size $|N(Y_i)| \leqslant 2^{d+1} + 2f(2^d) =: h$

and treedepth (and thus treewidth) at most d. Applying the protrusion reduction rule to each individual cluster as in Lemma 5 then yields an equivalent instance (G', ξ') with

$$|V(G')| = |Y_0| + \sum_{i=1}^{\ell} Y_i' \leqslant \mathcal{O}(|S|) + \ell \cdot \rho(d + 2^{d+1} + 2f(2^d), d) = \mathcal{O}(|S|) \cdot \mathcal{O}(1) = \mathcal{O}(|S|)$$

where Y_i' denote the clusters obtained through applications of the reduction rule. As G is degenerate, the above bound implies that $|V(G')| + |E(G')| = \mathcal{O}(|S|)$, too. □

Several graph problems have finite integer index on the class of all graphs and thus admit linear kernels on graphs of bounded expansion if parameterized by a treedepth modulator.

Corollary 3. *The following graph problems have finite integer index, and hence have linear kernels in graphs of bounded expansion, when the parameter is the size of a modulator to constant treedepth:* Dominating Set*,* r-Dominating Set*,* Efficient Dominating Set*,* Connected Dominating Set*,* Vertex Cover*,* Hamiltonian Path*,* Hamiltonian Cycle*,* Connected Vertex Cover*,* Independent Set*,* Feedback Vertex Set*,* Edge Dominating Set*,* Induced Matching*,* Chordal Vertex Deletion*,* Odd Cycle Transveral*,* Induced d-Degree Subgraph*,* Min Leaf Spanning Tree*,* Max Full Degree Spanning Tree*.*

For a more comprehensive list of problems that have FII in general graphs (and hence fall under the purview of the above corollary), see [6].

Some problems do not have FII in general (see [12]) but only when restricted to graphs of bounded treedepth, and for those we have the same conclusion in the following:

Lemma 6. *Let $\mathcal{G}$ be any graph class and $\mathcal{G}(d)$ be those graphs of $\mathcal{G}$ that have treedepth at most d. The problems* Longest Path*,* Longest Cycle*,* Exact s,t-Path*,* Exact Cycle *restricted to $\mathcal{G}$ have FII in $\mathcal{G}(d) \subseteq \mathcal{G}$ for any $d \in \mathbf{N}$.*

Proof. Let Π be any one of the mentioned problems restricted to $\mathcal{G}$, and let d, t be constants. Consider the class $\mathcal{G}_t$ of t-boundaried graphs over $\mathcal{G}$, and let $T = \{0, 1, \ldots, t\}$.

We define a *configuration* of Π with respect to $\mathcal{G}_t$ as a mutiset

$$C = \{(s_1, d_1, t_1), \ldots, (s_p, d_p, t_p)\}$$

of triples from $T \times \mathbf{N} \times T$. We say a t-boundaried graph $G \in \mathcal{G}_t$ *satisfies* the configuration C if there exists a set of (distinct) paths $P_1, \ldots, P_p$ in G such that

- s_i, t_i can only be endvertices of P_i, $V(P_i) \cap bd(G) \subseteq \{s_i, t_i\}$, and $|P_i| = d_i$, for $1 \leqslant i \leqslant p$,
- $V(P_i) \cap V(P_j) \subseteq bd(G)$ for $1 \leqslant i < j \leqslant p$,
- $V(P_i) \cap V(P_j) \cap V(P_k) = \emptyset$ for $1 \leqslant i < j < k \leqslant p$.

Note that, for simplicity, we identify the boundary vertices in $bd(G)$ with their labels $1,\dots,t$ from T. Moreover, s_i, t_i can take the value 0 which is not contained in $bd(G)$: semantically these tuples describe paths which intersect the boundary of G at only one or no vertex. Another special case are tuples with $s_i = t_i$ and $d = 0$: those describe single vertices of the boundary. In short, a graph satisfies a configuration if it contains internally non-intersecting paths of length and endvertices prescribed by the tuples of the configuration, and no three of the paths are prescribed to have the same endvertex (hence some configurations are not satisfiable at all, but this is a small technicality).

The *signature* $\sigma[G]$ of a graph $G \in \mathcal{G}_t$ is a function from the configurations into $\{0,1\}$ where $\sigma[G](C) = 1$ iff G satisfies C. We claim that the equivalence relation $\simeq_\sigma$ defined via

$$G_1 \simeq_\sigma G_2 \iff \sigma[G_1] \equiv \sigma[G_2] \text{ for } G_1, G_2 \in \mathcal{G}_t$$

is a refinement of $\equiv_{\Pi,t}$. We provide only a sketch for $\Pi =$ Longest Path, the proofs for the other problems work analogous. To this end we assume the contrary, that $\sigma[G_1] \equiv \sigma[G_2]$ while $G_1 \not\equiv_{\Pi,t} G_2$. Up to symmetry, this means that for all integers c there exists a graph $G_3 \in \mathcal{G}_t$ such that $(G_1 \oplus G_3, \ell) \in \Pi$ but $(G_2 \oplus G_3, \ell + c) \notin \Pi$. We choose $c = 0$ and show the contradiction. Thus the graph $G_1 \oplus G_3$ contains a path P of length ℓ but $G_2 \oplus G_3$ does not.

Using the implicit order given through the vertex order of P we sort the subpaths of P contained in $P \cap G_1$ and so obtain a sequence of paths $P_1,\dots,P_q \subseteq G_1$, each with at most two vertices – the ends, in $bd(G_1)$. By identifying each subpath P_i with the tuple (s_i, d_i, t_i) where $d_i = |P_i|$ and s_i is the label of the start of P_i in $bd(G_1)$ (or 0 if $s_i \notin bd(G_1)$) and t_i the label of the end of P_i in $bd(G_1)$ (ditto), we obtain a configuration $C_P = \{(s_1,d_1,t_1),\dots,(s_q,d_q,t_q)\}$. Now, G_1 satisfies C_P by the definition. Since $\sigma[G_1](C_P) = \sigma[G_2](C_P)$, there exists a set of paths $Q_1,\dots,Q_q \subseteq G_2$ witnessing that G_2 satisfies C_P. But then $Q_1,\dots,Q_q$ together with $P \cap G_3$ form a path Q of length ℓ in $G_2 \oplus G_3$, a contradiction.

Second, although $\simeq_\sigma$ is generally of infinite index, we claim that for every t, only a finite number of equivalence classes of $\simeq_\sigma$ carry a representative from $\mathcal{G}_t(d)$ – the subclass of treedepth at most d. This is rather easy since graphs of treedepth $\leqslant d$ do not contain paths of length $2^d - 1$ or longer, and so a graph $G \in \mathcal{G}_t(d)$ can satisfy a configuration $C = \{(s_1,d_1,t_1),\dots,(s_p,d_p,t_p)\}$ only if $d_i \in \{0,1,\dots,2^d-2\}$ for $1 \leqslant i \leqslant p$. Recall, each boundary vertex label occurs at most twice among $s_1,t_1,\dots,s_p,t_p$ in a satisfiable configuration. Hence only finitely many such configurations C can be satisfied by a graph from $\mathcal{G}_t(d)$, and consequently, finitely many function values of $\sigma[G]$ are nonzero for any $G \in \mathcal{G}_t(d)$ and the number of the nonempty classes of $\simeq_\sigma$ restricted to $\mathcal{G}_t(d)$ is finite. □

Lemma 7. *Let $\mathcal{G}$ be any graph class and $\mathcal{G}(w)$ be those graphs of $\mathcal{G}$ that have treewidth at most w. The problems* Treewidth *and* Pathwidth *restricted to $\mathcal{G}$ have FII in $\mathcal{G}(w) \subseteq \mathcal{G}$ for any $w \in \mathbf{N}$.*

Proof. Let $\Pi =$ Treewidth (the proof works analogously for Pathwidth) restricted to $\mathcal{G}$, and let w, t be constants. Consider the class $\mathcal{G}_t$ of t-boundaried graphs over $\mathcal{G}$, and let $U = \{1,2,\dots,t\}$. We again, for simplicity, identify the boundary vertices in a graph from $\mathcal{G}_t$ with their labels $1,\dots,t$ from U.

We mimic the proof of Lemma 6 with some changes. We define a *configuration* of Π wrt. $\mathcal{G}_t$ as a set $C = \{(\mathcal{X}_1, w_1), \ldots, (\mathcal{X}_p, w_p)\}$ of pairs such that $\mathcal{X}_i \subseteq 2^U$ and $w_i \in \mathbf{N}$ for $i = 1, \ldots, p$. We say a t-boundaried graph $G \in \mathcal{G}_t$ *satisfies* the configuration C if there exists a collection of induced subgraphs $H_1, \ldots, H_p$ of G such that

- $V(H_i) \cap V(H_j) \subseteq bd(G)$ for $1 \leqslant i < j \leqslant p$, and $H_1 \cup \ldots \cup H_p = G$,
- there exists a tree decomposition $(T_i, \mathcal{W}_i)$, $i = 1, 2, \ldots, p$, of the graph H_i of width at most w_i,
- each $X \in \mathcal{X}_i$ is a bag in this decomposition, i.e., $X \in \mathcal{W}_i$.

The *signature* $\sigma[G]$ of a graph $G \in \mathcal{G}_t$ is a function from the configurations into $\{0, 1\}$ where $\sigma[G](C) = 1$ iff G satisfies C. We claim that the equivalence relation $\simeq_\sigma$ defined via

$$G_1 \simeq_\sigma G_2 \iff \sigma[G_1] \equiv \sigma[G_2] \text{ for } G_1, G_2 \in \mathcal{G}_t$$

is a refinement of $\equiv_{\Pi,t}$. To this end we assume the contrary, that $\sigma[G_1] \equiv \sigma[G_2]$ while $G_1 \not\equiv_{\Pi,t} G_2$. Up to symmetry, this means that for all integers c there exists a graph $G_3 \in \mathcal{G}_t$ such that $(G_1 \oplus G_3, k) \in \Pi$ but $(G_2 \oplus G_3, k + c) \notin \Pi$. We choose $c = 0$ and show the contradiction. Thus the graph $G_1 \oplus G_3$ has a tree decomposition $(T, \mathcal{W})$ of width k but $G_2 \oplus G_3$ does not. We will set $B = bd(G_1)$ and assume for simplicity that $B = bd(G_2) = bd(G_3) = U$, i.e., $B \subseteq G_1 \oplus G_3$ as well as $B \subseteq G_2 \oplus G_3$. As B is a vertex-separator of $G_1 \oplus G_3$, we can assume that no bag in $\mathcal{W}$ contains both vertices from $G_1 \setminus B$ and from $G_3 \setminus B$. We will further assume that each bag in $\mathcal{W}$ appears exactly once in the tree decomposition, that every subset $X \subseteq B$ which is contained in some bag also exists exclusively as a bag $X \in \mathcal{W}$, and that for no adjacent bags their union contains both vertices from $G_1 \setminus B$ and from $G_3 \setminus B$ (all three conditions can easily be enforced without increasing the width of the decomposition).

We color the nodes of T with colors white, black and red according to the following criterion: every $x \in V(T)$ is assigned the color $c(x)$, where $c(x)$ is

- red if $W_x \subseteq B$, and otherwise
- white if $W_x \subseteq V(G_1)$ and black if $W_x \subseteq V(G_3)$.

The above conditions on the structure of $(T, \mathcal{W})$ now imply that c partitions the nodes $V(T)$ into $T_{white}, T_{black}, T_{red}$, and that no white node is adjacent to a black node in T.

From this coloring we create a collection of subtrees $T_1, \ldots, T_q$ – the connected components of $T - T_{black}$. Let H_i, $i = 1, \ldots, q$, be the subgraph of G_1 induced by $\bigcup_{x \in V(T_i)} W_x$, and let $(T_i, \mathcal{W}_i)$ denote the corresponding tree decomposition of H_i. We denote by w_i the width of $(T_i, \mathcal{W}_i)$ and set $\mathcal{X}_i = \{W_x \in \mathcal{W}_i : x \in V(T_{red})\}$. Now, the subgraphs $H_1, \ldots, H_q$ witness that the graph G_1 satisfies the configuration $C_T = \{(\mathcal{X}_1, w_1), \ldots, (\mathcal{X}_q, w_q)\}$ by definition.

Since $\sigma[G_1](C_T) = \sigma[G_2](C_T)$, there exists a collection of induced subgraphs $H'_1, \ldots, H'_q$ of G_2, and their tree decompositions $(T'_1, \mathcal{W}'_1), \ldots, (T'_q, \mathcal{W}'_q)$ witnessing that also G_2 satisfies C_T (particularly with the same widths $w_1, \ldots, w_q$, respectively). Moreover, for each $x \in$

$V(T_{red})$, the bag $W'_x \in \mathcal{W}'_i$ (for the appropriate i such that $W_x \in \mathcal{W}_i$ above) is the same as $W'_x = W_x \in \mathcal{W}$ on the boundary B. We make T' as the union, by identification of nodes in $V(T_{red})$, of $T - V(T_{white})$ with $T'_1 \cup \ldots \cup T'_q$, and set $\mathcal{W}'$ to be the union of $\mathcal{W}$ restricted to the nodes of $T_{black} \cup T_{red}$ with $\mathcal{W}'_1 \cup \ldots \cup \mathcal{W}'_q$. But then $(T', \mathcal{W}')$ is a tree decomposition of width k in $G_2 \oplus G_3$, a contradiction.

Second, although $\simeq_\sigma$ is generally of infinite index, we claim that for every t, only a finite number of equivalence classes of $\simeq_\sigma$ carry a representative from $\mathcal{G}_t(w)$ – the subclass of treewidth at most w. For this we claim that a graph G of treewidth $\leqslant w$ can satisfy a configuration $C = \{(\mathcal{X}_1, w_1), \ldots, (\mathcal{X}_p, w_p)\}$ only if G satisfies also the configuration $\{(\mathcal{X}_1, w'_1), \ldots, (\mathcal{X}_p, w'_p)\}$ where $w'_i = \min(w_i, w + t)$ for $1 \leqslant i \leqslant p$. To see this, notice that one can take a tree decomposition of whole G restricted to witness subgraphs H_i (notation as above) and add suitable subsets of the boundary to (some) bags, to form the witness tree decomposition for $(\mathcal{X}_i, w'_i)$. Moreover, $p \leqslant 2^{2^t}$ as every combination of subsets of the boundary can appear at most once. Therefore, finiteness of $\simeq_\sigma$ restricted to $\mathcal{G}_t(w)$ follows as at the end of Lemma 6. □

Corollary 4. *The problems* Longest Path*,* Longest Cycle*,* Exact s,t-Path*,* Exact Path*,* Treewidth*, and* Pathwidth *have linear kernels in graphs of bounded expansion with the size of a modulator to constant treedepth as the parameter.*

4.1 Extension to larger graph classes

We can extend our result to classes of graphs of *locally bounded expansion* and furthermore to graphs that are *nowhere dense.*

Definition 12 (Locally bounded expansion [16])**.** A graph class $\mathcal{G}$ has *locally bounded expansion* if there exists a function $f\colon \mathbf{N} \times \mathbf{N} \to \mathbf{R}$ (called the *expansion function*) such that for every graph $G \in \mathcal{G}$ and all $r, d \in \mathbf{N}$ and every vertex $v \in V(G)$, it holds that $\nabla_r(G[N_d(v)]) \leqslant f(d, r)$.

Definition 13 (Nowhere dense [27,28])**.** A graph class $\mathcal{G}$ is *nowhere dense* if for all $r \in \mathbf{N}$ it holds that $\omega(\mathcal{G} \mathbin{\triangledown} r) < \infty$.

In the above definition we use the natural extension of ω to classes of graphs via $\omega(\mathcal{G}) = \sup_{G \in \mathcal{G}} \omega(G)$. Note that both graph classes are closed under taking shallow minors in the sense that $\mathcal{G} \mathbin{\triangledown} r$ has locally bounded expansion (is nowhere dense) if $\mathcal{G}$ has locally bounded expansion (is nowhere dense), albeit with a different expansion function (a different bound on the clique size of r-shallow minors).

We claim the following two kernelization result for the above classes, which in particular apply to all problems listed in Section 4.

Theorem 2. *Let $\mathcal{G}$ be a graph class of locally bounded expansion and for $p \in \mathbf{N}$, let $\mathcal{G}(p) \subseteq \mathcal{G}$ be the subclass of graphs of treedepth at most p. Let $\Pi_{\mathcal{G}}$ be a graph problem that has finite integer index on $\mathcal{G}(p)$ for all $p \in \mathbf{N}$ and let $d \in \mathbf{N}$ be a constant. Then there is an algorithm that takes as input $(G, \xi) \in \Pi_{\mathcal{G}}$ and, in polynomial time, outputs an equivalent instance (G', ξ') such that $|G'| = O(|S|^2)$, where S is an optimum treedepth-d modulator of the graph G.*

Theorem 3. *Let $\mathcal{G}$ be a nowhere-dense graph class for $p \in \mathbf{N}$, let $\mathcal{G}(p) \subseteq \mathcal{G}$ be the subclass of graphs of treedepth at most p. Let $\Pi_{\mathcal{G}}$ be a graph problem that has finite integer index on $\mathcal{G}(p)$ for all $p \in \mathbf{N}$ and let $d \in \mathbf{N}$ be a constant. Then there is an algorithm that takes as input $(G, \xi) \in \Pi_{\mathcal{G}}$ and, in polynomial time, outputs an equivalent instance (G', ξ') such that $|G'| = \mathcal{O}(|S|^c)$ for some constant c, where S is an optimum treedepth-d modulator of the graph G.*

The proofs of Theorems 2 and 3 follow analogously to the proof of Theorem 1 using Lemma 8 (see below) in place of Lemma 3. We need additional notation. Let $\#\omega(G)$ denote the number of complete subgraphs of G. For a graph class $\mathcal{G}$ and an integer ℓ we let $\mathcal{G}_{\leqslant \ell} := \{H \in \mathcal{G} \mid |H| \leqslant \ell\}$ denote those graphs of $\mathcal{G}$ which have at most ℓ vertices.

Definition 14 (Greatest reduced average clique density)**.** For a graph G and integer r we define $\square_r(G) = \max_{H \in G \triangledown r}(\#\omega(H)/|H|)$ to be the *greatest reduced clique density* (clique-grad) with rank r of G. For a graph class $\mathcal{G}$ the *clique expansion* with rank r is defined as $\square_r(\mathcal{G}) = \sup_{G \in \mathcal{G}} \square_r(G)$.

Lemma 8. *Let $G = (X, Y, E)$ be a bipartite graph let $\mathcal{H}_X = (G \triangledown 1)_{\leqslant |X|}$. Then there are at most*

1. *$2\nabla_0(\mathcal{H}_X) \cdot |X|$ vertices in Y with degree larger than $\omega(\mathcal{H}_X)$;*
2. *$\Big(\square_0(\mathcal{H}_X) + 2\nabla_0(\mathcal{H}_X)\Big) \cdot |X|$ subsets $X' \subseteq X$ such that $X' = N(u)$ for some $u \in Y$.*

Proof. We construct a sequence of graphs $G_0, G_1, \ldots, G_\ell$ analogous to the proof of Lemma 3 Note that, by construction, we have that $G_i[X] \in \mathcal{H}_X$ for $1 \leqslant i \leqslant \ell$. In particular, this implies that $G_\ell[X]$ has at most $2\nabla_0(\mathcal{H}_X) \cdot |X|$ edges.

Let us now prove the first claim. To this end, assume that there is a vertex $v \in Y \cap V(G_\ell)$ such that $\deg(v) > \omega(\mathcal{H}_X)$. We claim that $G_\ell[N(v)]$ (where $N(v) \subseteq X$) is a clique. If not, we could choose a pair of non-adjacent vertices in $G_\ell[N(v)]$ and construct a $(\ell+1)$-th graph for the sequence which would contradict the fact that G_ℓ is the last graph of the sequence. However, the set $N(v)$ then induces a clique of size larger than $\omega(\mathcal{H}_X)$, a contradiction.

Hence we conclude that no vertex of $Y \cap V(G_\ell)$ has degree $> \omega(\mathcal{H}_X)$ in G_ℓ (and thus in G). Therefore the vertices of Y of degree $> \omega(\mathcal{H}_X)$ in the graph G, if there were any, must have been deleted during the edge contractions that resulted in the graph G_ℓ. As every contraction added at least one edge between vertices in X and since $G_\ell[X]$ contains at most $2\nabla_0(\mathcal{H}_X) \cdot |X|$ edges, the first claim follows.

For the second claim, consider the set $Y' = Y \cap V(G_\ell)$. As observed above, the neighborhood of every vertex $v \in Y'$ induces a clique in $G_\ell[X]$. The number such sets therefore can be upper bounded by the number of cliques in $G_\ell[X]$, which in turn can be bounded as follows:

$$\#\omega(G_\ell[X]) = \square_0(G_\ell[X])|X| \leqslant \square_0((G \triangledown 1)_{\leqslant |X|})|X| = \square_0(\mathcal{H}_X)|X|$$

In total then the number of subsets of X that are neighborhoods of vertices in Y in G is at most $(\square_0(\mathcal{H}_X) + 2\nabla_0(\mathcal{H}_X))|X|$, where we accounted for vertices of Y lost via contractions by the bound on the number of edges in $G_\ell[X]$. □

The following two corollaries are analogs of Corollary 1 and 2 and will be used in a similar fashion.

Corollary 5. *Let $\mathcal{G}$ be a graph-class. Suppose that for $G \in \mathcal{G}$ and $S \subseteq V(G)$, $C_1, \ldots, C_s$ are disjoint connected subgraphs of $G - S$ satisfying the following two conditions: for $1 \leqslant i \leqslant s$, $diam(G[V(C_i)]) \leqslant \delta$ and $|N_S(C_i)| > \omega(\mathcal{H}_S)$ where $\mathcal{H}_S = (G \mathbin{\triangledown} (\delta+1))_{\leqslant |S|}$. Then $s \leqslant 2\nabla_0(\mathcal{H}_S) \cdot |S|$.*

Proof. We construct an auxiliary bipartite graph $\tilde{G}$ with partite sets S and $Y = \{C_1, \ldots, C_s\}$. There is an edge between C_i and $x \in S$ iff $x \in N_S(C_i)$. Note that $\tilde{G}$ is a shallow minor at depth δ of G by the assumption, and therefore $(\tilde{G} \mathbin{\triangledown} 1)_{\leqslant |S|} \subseteq \mathcal{H}_S$. By Lemma 8,

$$s \leqslant 2\nabla_0((\tilde{G} \mathbin{\triangledown} 1)_{\leqslant |S|})|S| \leqslant 2\nabla_0(\mathcal{H}_S)|S|.$$

□

Corollary 6. *Let $\mathcal{G}$ be a graph-class. Suppose that for $G \in \mathcal{G}$ and $S \subseteq V(G)$, $\mathcal{C}_1, \ldots, \mathcal{C}_t$ are sets of connected components of $G - S$ such that for all $C, C' \in \bigcup_i \mathcal{C}_i$ it holds that $C, C' \in \mathcal{C}_j$ for some j if and only if $N_S(C) = N_S(C')$. Let $\delta > 0$ be a bound on the diameter of the components, i.e. for all $C \in \bigcup_i \mathcal{C}_i$, $diam(G[V(C)]) \leqslant \delta$. Then there can be only at most $t \leqslant (\Box_0(\mathcal{H}_S) + 2\nabla_0(\mathcal{H}_S)) \cdot |S|$ such sets $\mathcal{C}_i$ where again $\mathcal{H}_S = (G \mathbin{\triangledown} (\delta + 1))_{\leqslant |S|}$.*

Proof. As in the proof of Corollary 5, we construct a bipartite graph $\tilde{G}$ with partite sets S and $Y = \{C_1, \ldots, C_r\}$, where the vertices C_j represent connected components in $\bigcup_i \mathcal{C}_i$ and C_j has an edge to $x \in S$ iff $x \in N_S(C_j)$. As before, $\tilde{G}$ is a shallow minor at depth δ of G and therefore $(\tilde{G} \mathbin{\triangledown} 1)_{\leqslant |S|} \subseteq \mathcal{H}_S$. By Lemma 8,

$$\begin{aligned} t &\leqslant |\{S' \subseteq S \mid \exists C_i \in Y : N(C_i) = S'\}| \\ &\leqslant (\Box_0((\tilde{G} \mathbin{\triangledown} 1)_{\leqslant |S|}) + 2\nabla_0((\tilde{G} \mathbin{\triangledown} 1)_{\leqslant |S|})) \cdot |S| \\ &\leqslant (\Box_0(\mathcal{H}_S) + 2\nabla_0(\mathcal{H}_S)) \cdot |S|. \end{aligned}$$

□

Note that, using the notation of Lemma 8, we have the trivial bounds $2\nabla_0(\mathcal{H}_X) \leqslant |X|$ and $\Box_0(\mathcal{H}_X) \leqslant |X|^{\omega(\mathcal{H}_X)-1}$. For graphs of locally bounded expansion this second bound can be improved as follows.

Lemma 9. *Let $\mathcal{G}$ be a graph class with local expansion bounded by $f\colon \mathbf{N} \times \mathbf{N} \to \mathbf{R}$. Then for any graph $G \in \mathcal{G}$, any constant c and any integer $0 \leqslant \ell \leqslant |V(G)|$, $\Box_0\Big((G \mathbin{\triangledown} c)_{\leqslant \ell}\Big) \leqslant 4^{f(1+c,0)}\ell$.*

Proof. Consider any $H \in (G \mathbin{\triangledown} c)_{\leqslant \ell}$. Note that $H \in (G \mathbin{\triangledown} c)_{\leqslant \ell} \subseteq G \mathbin{\triangledown} c$, and thus H has local expansion bounded by $f'(d, r) = f(d + c, r)$.

We upper-bound the cliques in H iteratively as follows: pick a vertex v, count all cliques that contain v and add those to the number of cliques in $H - v$. Now, all cliques that contain a fixed vertex v must be contained in $N[v]$. As $G[N[v]]$ is a radius-one subgraph of H, it has bounded expansion with expansion function $f'(1, r) = f(1 + c, r)$ and thus is

$2f(1+c,0)$-degenerate. We can now apply the result of [32], stating that every d-degenerate graph G with $n \geqslant d$ vertices has at most $2^d(n-d+1)$ cliques. Doing so we see that $G[N[v]]$ contains at most $2^{2f(1+c,0)}|N[v]| \leqslant 4^{f(1+c,0)}|H| \leqslant 4^{f(1+c,0)}\ell$ cliques. Iterating this counting over all vertices of H then yields a generous bound of $4^{f(1+c,0)}\ell^2$ and therefore we obtain the desired bound for the clique density through division by ℓ. □

The following generalization of Lemma 4 follows easily using the above two corollaries.

Lemma 10. *Let $\mathcal{G}$ be a graph class, $G \in \mathcal{G}$ and $S \subseteq V(G)$ be a set of vertices such that $\mathbf{td}(G-S) \leqslant d$ (d a constant). Let $\mathcal{H}_S = (G \triangledown 2^d)_{\leqslant |S|}$. If $\omega(\mathcal{H}_S)$ is a constant, then there is an algorithm that runs in time linear in $|G|$ and partitions $V(G)$ into sets $Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ such that the following hold:*

1. *$S \subseteq Y_0$ and $|Y_0| \leqslant 2\nabla_0(\mathcal{H}_S) \cdot |S|$;*
2. *for $1 \leqslant i \leqslant \ell$, Y_i induces a set of connected components of $G - Y_0$ that have the same neighborhood in Y_0 of size at most $\omega(\mathcal{H}_S)$;*
3. *$\ell \leqslant \Big(\square_0(\mathcal{H}_S) + 2\nabla_0(\mathcal{H}_S)\Big) \cdot |S|$.*

Proof. We proceed exactly as in the proof of Lemma 4 using $t := \omega(\mathcal{H}_S)$ and the bounds from Corollary 5 and 6 □

We are now ready to prove the two theorems.

Proof of Theorem 2. Analogously to the proof of Theorem 1 we use Lemma 10 to obtain a protrusion-decomposition $Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ in place of Lemma 4. Let G be a graph from a class of locally bounded expansion and let d be an integer and $S \subset V(G)$ be a treedepth-d modulator of G. It is left to show that for $\mathcal{H}_S = (G \triangledown 2^d)_{\leqslant |S|}$ the bounds of Lemma 10 are indeed quadratic in $|S|$. Clearly, $\nabla_0(G) \leqslant |G|$, thus $\nabla_0(\mathcal{H}_S) \leqslant |S|$ and therefore $|Y_0| = \mathcal{O}(|S|^2)$. The bound $\square_0(\mathcal{H}_S) = \mathcal{O}(|S|)$ was proved in Lemma 9 and therefore $\ell \leqslant (\square_0(\mathcal{H}_S) + 2\nabla_0(\mathcal{H}_S))|S| = \mathcal{O}(|S|^2)$ and the claim follows. □

Proof of Theorem 3. Analogously to the proof of Theorem 1 we use Lemma 10 to obtain a protrusion-decomposition $Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ in place of Lemma 4. Let G be a graph from a nowhere-dense graph class and let d be an integer and $S \subset V(G)$ a treedepth-d modulator of G. It is left to show that for $\mathcal{H}_S = (G \triangledown 2^d)_{\leqslant |S|}$ the bounds of Lemma 10 are indeed polynomial in $|S|$. Clearly, $\nabla_0(G) \leqslant |G|$, thus $\nabla_0(\mathcal{H}_S) \leqslant |S|$ and therefore $|Y_0| = \mathcal{O}(|S|^2)$.

For $\square_0(\mathcal{H}_S)$ we use the trivial bound of $\square_0(\mathcal{H}_S) \leqslant |S|^{\omega(\mathcal{H}_S)-1}$, so it is left to show that $\omega(\mathcal{H}_S)$ is a constant. As $\mathcal{H}_S \subseteq G \triangledown 2^d$ and per definition of nowhere-dense graph classes, $\omega(G \triangledown r) < \infty$ for every constant r, the claim follows. □

5 Structural Parameterizations of Longest Path

In this section we show that the problem Longest Path has a polynomial kernel when parameterized by a modulator to constant treedepth. Our result almost entirely closes the gap between the polynomial kernel of Longest Path when parameterized by the size of a vertex cover and the no polynomial kernel result for Longest Path when parameterized by the size of a modulator to pathwidth two [7].

It is well-known that Longest Path can be solved in linear time if the treewidth of the input graph is bounded by some constant [4]. Because of the relationship between treewidth and treedepth (see Section 2) this result carries over to treedepth.

Proposition 5. Longest Path *can be solved in linear time if the treedepth of the input graph is bounded by some constant.*

The following lemma is at the very core of our result.

Lemma 11. *For fixed $d \in \mathbf{N}, d \geqslant 1$, let $S \subseteq V(G)$ be a treedepth-d modulator of a graph G and let $k = |S|$. Then there is an induced subgraph G' of G and a set $S' \subseteq V(G')$ such that: (1) G and G' are equivalent instances of* Longest Path *(for the same path length), (2) G' and S' can be computed from G and S in time $\mathcal{O}(k^2 \cdot |V(G)|)$, and (3) S' is a treedepth-$(d-1)$ modulator of G' of size $|S'| \leqslant (k+1)^3$.*

Proof. Let $\mathcal{U}$ be the family of vertex sets of all connected components of $G - S$. Since for each $U \in \mathcal{U}$ the graph $G[U]$ has treedepth at most d, there exists $r_U \in U$ (the root of some treedepth d decomposition) such that $G[U - \{r_U\}]$ has treedepth $d - 1$. Therefore if we can find in time $\mathcal{O}(k^2 \cdot |V(G)|)$ a subfamily $\mathcal{U}' \subseteq \mathcal{U}$ of size at most $(k+1)^3 - k$ such that $G' = G[S \cup \bigcup_{U \in \mathcal{U}'} U]$ is an equivalent instance of Longest Path, the claim of the lemma follows. To see this, notice that we can use vertices r_U, one for each $U \in \mathcal{U}'$, together with vertices from S to form treedepth-$(d-1)$ modulator S' of G'. The modulator S' will therefore consist of k vertices from S and at most $(k+1)^3 - k$ new vertices, one from each component of $\mathcal{U}'$, and so $|S'| \leqslant (k+1)^3$, as claimed.

In the rest of the proof, we show that we can find the family $\mathcal{U}'$ with the aforementioned properties in desired time.

Assume $|\mathcal{U}| > k + 1$. For all $U \in \mathcal{U}$ and $x, y \in S$ with $x \neq y$ we denote

i. by $\mathrm{LP}(U)$ a longest path in the graph $G[U]$ (we choose any one if not unique), and by $U_0 \in \mathcal{U}$ a representative achieving maximum value $|\mathrm{LP}(U_0)|$ over $\mathcal{U}$;

ii. by $\mathrm{LP}(x, U)$ a longest path starting from x in the graph $G[\{x\} \cup U]$, and by $\mathcal{U}_x \subseteq \mathcal{U}$ a subfamily of $|\mathcal{U}_x| = k + 1$ ("top $k + 1$ representatives" by $|\mathrm{LP}(x, U)|$) such that for any $U_1 \in \mathcal{U}_x$, $U_2 \in \mathcal{U} \setminus \mathcal{U}_x$ it is $|\mathrm{LP}(x, U_1)| \geqslant |\mathrm{LP}(x, U_2)|$;

iii. by $\mathrm{LP}(x, y, U)$ a longest path between x and y in the graph $G[\{x, y\} \cup U]$, or $\mathrm{LP}(x, y, U) = \emptyset$ if no such path exists, and analogously by $\mathcal{U}_{x,y} \subseteq \mathcal{U}$ a subfamily of $|\mathcal{U}_{x,y}| = k + 1$ ("top $k + 1$ representatives" by $|\mathrm{LP}(x, y, U)|$) such that for any $U_1 \in \mathcal{U}_{x,y}$, $U_2 \in \mathcal{U} \setminus \mathcal{U}_{x,y}$ it is $|\mathrm{LP}(x, y, U_1)| \geqslant |\mathrm{LP}(x, y, U_2)|$.

Because $\mathbf{td}(G[U]) \leqslant \mathbf{td}(G[\{x\} \cup U]) \leqslant \mathbf{td}(G[\{x,y\} \cup U]) \leqslant d+2$ (a constant), it follows from Proposition 5 that $\mathrm{LP}(U)$, $\mathrm{LP}(x,U)$, $\mathrm{LP}(x,y,U)$ can each be computed in linear time, and hence the whole computation of $U_0, \mathcal{U}_x, \mathcal{U}_{x,y}$ can be done in $\mathcal{O}(k^2 \cdot |V(G)|)$ time.

We claim that the family $\mathcal{U}' = \{U_0\} \cup \bigcup_{x \in S} \mathcal{U}_x \cup \bigcup_{x,y \in S, x \neq y} \mathcal{U}_{x,y}$ together with S induces graph G' which satisfies the conclusion of the lemma. Clearly, $|\mathcal{U}'| \leqslant \binom{k}{2}(k+1) + k(k+1) + 1 = \frac{1}{2}k(k+1)^2 + 1 \leqslant (k+1)^3$. It remains to show that if G has a path of length at least ℓ then so does $G' = G[S \cup \bigcup_{U \in \mathcal{U}'} U]$.

Let P be a path of length at least ℓ in G and let $q = |V(P) \cap S| \leqslant k$. Then S "cuts" P into $q+1$ sections, i.e., we can write $P = P_0 \cup P_1 \cup \ldots \cup P_q$ where P_i, $i = 0, \ldots, q$ are mutually edge-disjoint paths disjoint from S except possibly at their ends. Suppose that $P \not\subseteq G'$. There are three cases to consider for the subpaths P_i:

I. $q = 0$ and $P = P_0$. Then the length of P is at most $|\mathrm{LP}(U_0)|$ by the definition, and hence we can choose $P' := \mathrm{LP}(U_0) \subseteq G'$ straight away.

II. $q \geqslant 1$ and $P_0 \not\subseteq G'$ or $P_q \not\subseteq G'$. Consider, without loss of generality, the latter case $P_q \not\subseteq G'$ and let $\{x\} = V(P_q) \cap S$. Then the length of P_q is at most $|\mathrm{LP}(x,U)|$ for any $U \in \mathcal{U}_x$ by the definition. Notice that each of the $q \leqslant k$ paths P_i, $i = 0, \ldots, q-1$, can intersect only at most one component from $\mathcal{U}$ by connectivity (and P_q is disjoint from all of $\mathcal{U}_x$). Hence, at least $k+1-q \geqslant 1$ component(s) in $\mathcal{U}_x$, say U_1, is disjoint from whole P. Then in P we replace P_q with $\mathrm{LP}(x, U_1)$.

III. $q \geqslant 1$ and $P_i \not\subseteq G'$ where $0 < i < q$. Let $\{x,y\} = V(P_i) \cap S$. Then the length of P_i is at most $|\mathrm{LP}(x,y,U)|$ for any $U \in \mathcal{U}_{x,y}$ by the definition. For the same reason as above there exists a component $U_2 \in \mathcal{U}_{x,y}$ not intersected by P, and we then in P replace P_i with $\mathrm{LP}(x,y,U_2)$.

Repeating II, III for all sections of P, the resulting path $P' \subseteq G'$ has length at least $|P| \geqslant \ell$, and this concludes the proof of the lemma. □

Theorem 4. *Let $d \in \mathbf{N}$ be a constant, and let the function g be defined as follows; $g(0,k) = k$ and $g(i,k) = g\Big(i-1, (k+1)^3\Big)$. Then* Longest Path *has a* polynomial kernel *of size at most $g(d,k)$ parameterized by the size k of a modulator to treedepth d where, asymptotically, $g(d,k) = O(k^{3^d})$. This kernel is computable in time $\mathcal{O}(k^2 \cdot |V(G)|)$.*

Proof. Let G be a graph, and $S \subseteq V(G)$ a treedepth-d modulator of G. We proceed by induction on $d \geqslant 0$: For $d = 0$ we necessarily have $S = V(G)$ (cf. Lemma 11) and hence immediately a kernel of size $k = g(0,k)$. For $d > 0$, we apply Lemma 11 to obtain an equivalent instance G' with modulator S' of size $k' = |S'| \leqslant (k+1)^3$. Then G' can be kernelized to an instance of size at most $g(d-1,k')$ by the inductive assumption, and $g(d-1,k') \leqslant g(d,k)$ as desired. □

6 On the Ecology of Structural Parameters

A primary goal of parameterized complexity is to study how different parameters affect the complexity of classical problems. In particular, one aims to discover the boundaries of tractability by finding the weakest parameterization for which a problem is in FPT or admits polynomial kernels. It also provides further insight into what exactly does make hard problems hard. This study of how different parameters influence fixed-parameter tractability or polynomial kernelizability is referred to as *parameterized ecology* [17]. A surge in the interest of parameterized ecology has helped to make headway in the parameterized ecology program (see [8, 23, 24]).

In the quest for polynomial kernels one often has to consider two types of structural restrictions: restrictions on the input instances and restrictions on the parameters. Some problems are way too difficult in general to be tractable. For instance, Dominating Set is W[2]-complete and Independent Set is W[1]-complete, but both problems admit linear kernels on planar graphs [1]. This raises the question of whether these problems are tractable in more general graph classes under *stronger* parameterizations[4] such as e.g. the vertex cover number. It turns out that both these problems are indeed in FPT when parameterized by the vertex cover number. Other problems exist that seem to be much harder, such as Dominating Set, which does not admit a polynomial kernel even when parameterized by the solution size *and* the vertex cover number [14]. It is clear that if we wish to identify the boundary of polynomial kernelizability for problems that are as difficult as Dominating Set, we must necessarily restrict ourselves to special graph classes.

Another illustrative example is Longest Path. The standard parameterized version of this problem is in FPT in general graphs but has no polynomial kernel [2, 5]. When parameterized by the size of a vertex cover, it admits a quadratic kernel [7]. This leads us to the question as to whether there exist parameters weaker than vertex cover for which Longest Path has a polynomial kernel. One possibility is to use the treewidth as the parameter. But as was implicitly shown in [7], Longest Path does not admit a polynomial kernel even when parameterized by a modulator to a graph of pathwidth two.

In fact, we conjecture that Longest Path does not have a polynomial kernel in general graphs with respect to a modulator to a single path. Therefore if we want a polynomial kernel for this problem with respect to a parameter that modulates some graph property, then it seems that this property must not admit long paths. This is one reason why we chose to parameterize problems by the size of a modulator to bounded treedepth, as graphs of bounded treedepth have a bound on the longest path. In this case, we do indeed have a polynomial kernel, as was shown in Section 5. If we restrict the input instances by requiring that they are members of a graph class of bounded expansion then we obtain a linear kernel. On general graphs, the degree of the polynomial is a function of the treedepth. It is an interesting question whether this dependency can be removed.

[4] We use adjectives such as "strong" in the sense that they impose a greater structural restriction on the input instance. As a case in point, the vertex cover number of a graph is a stronger parameter than say, the feedback vertex number, since in the former case the "rest of the graph" is an independent set, whereas in the latter case it is a forest.

At first glance, a modulator to bounded treedepth seems to be a severely restricting parameter. But note that a vertex cover is a modulator to a treedepth-1 graph and hence our parameter is certainly less restrictive than the vertex cover number. If our conjecture for Longest Path holds, this is essentially the best parameter that we can hope for if we want a polynomial kernel. But the connection between treedepth and graphs of bounded expansion is deeper. Dvořák and Král showed that for any graph class $\mathcal{G}$ of bounded expansion and any positive integer p, there exists $q \in \mathbf{N}$ such that every graph $G \in \mathcal{G}$ has a vertex coloring with q colors such that for any i color classes, $1 \leqslant i \leqslant p$, induce a subgraph of treedepth at most i [26]. That is, any $G \in \mathcal{G}$ can be partitioned into a constant number of subgraphs each of constant treedepth.

Graph class	*Parameter*		
	natural	**tw**-*modulator*	**td**-*modulator*
	Dominating Set		
General graphs	W[2]	W[2]	no poly[a]
Nowhere dense	?	?	$\mathcal{O}(k^c)$
Locally bnd. exp.	?	?	$\mathcal{O}(k^2)$
Bounded expansion	?	?	$\mathcal{O}(k)$
Top. H-minor-free	$\mathcal{O}(k)$	$\mathcal{O}(k)$	$\mathcal{O}(k)$
H-minor-free	$\mathcal{O}(k)$	$\mathcal{O}(k)$	$\mathcal{O}(k)$
Planar	$\mathcal{O}(k)$	$\mathcal{O}(k)$	$\mathcal{O}(k)$
	Longest Path		
General graphs	no poly	no poly[b]	$\mathcal{O}(k^{3^d})$
Nowhere dense	no poly	no poly[b]	$\mathcal{O}(k^c)$
Locally bnd. exp.	no poly	no poly[b]	$\mathcal{O}(k^2)$
Bounded expansion	no poly	no poly[b]	$\mathcal{O}(k)$
Top. H-minor-free	no poly	?	$\mathcal{O}(k)$
H-minor-free	no poly	?	$\mathcal{O}(k)$
Planar	no poly	?	$\mathcal{O}(k)$
	Treewidth-t Vertex Deletion		
General graphs	$\mathcal{O}(k^{f(t)})$	no poly[c]	?
Nowhere dense	$\mathcal{O}(k^{f(t)})$	?	$\mathcal{O}(k^c)$
Locally bnd. exp.	$\mathcal{O}(k^{f(t)})$	?	$\mathcal{O}(k^2)$
Bounded expansion	$\mathcal{O}(k^{f(t)})$	?	$\mathcal{O}(k)$
Top. H-minor-free	$\mathcal{O}(k)$	$\mathcal{O}(k)$	$\mathcal{O}(k)$
H-minor-free	$\mathcal{O}(k)$	$\mathcal{O}(k)$	$\mathcal{O}(k)$
Planar	$\mathcal{O}(k)$	$\mathcal{O}(k)$	$\mathcal{O}(k)$

Figure 2: Overview of known kernelization results for selected problems on sparse graph classes. The gray fields highlight results from this paper.

The existence of a polynomial kernel is not the only relevant question. It is desirable for the kernel to be as small as possible. For problems with FII on graphs of bounded treedepth, we have shown the existence of a linear kernel on graphs of bounded expansion, which is obviously the best we can hope for. But is there a weaker parameter that still allows one to obtain linear kernels on graphs of bounded expansion for the same set of problems? One possibility is to use the size of a modulator to bounded treewidth as parameter. This is not likely to yield linear kernels for the following reason: Firstly, *any* graph class $\mathcal{G}$ can be transformed into a graph class $\tilde{\mathcal{G}}$ of bounded expansion by replacing every $G \in \mathcal{G}$ by $\tilde{G}$ obtained by replacing each edge in G by a path on $|V(G)|$ vertices. It is easy to verify that the operation of subdividing edges does *not* change the treewidth. Now let us consider Treewidth-t Vertex Deletion

[a] even if parameterized by the solution size plus the size of a minimal vertex cover

[b] even if parameterized by a modulator to pathwidth-2

[c] Assuming that $d > t$

which is the prototypical problem parameterized by a modulator to constant treewidth. An input to this problem consists of a graph G and parameter k. The question is whether there exist at most k vertices whose deletion from G results in a subgraph of treewidth at most t. Two special cases of this problem are Vertex Cover, where $t = 0$, and Feedback Vertex Set, where $t = 1$. It is well-known that Vertex Cover has an $\mathcal{O}(k)$ vertex-kernel [11] and that the best-known kernel for Feedback Vertex Set has $\mathcal{O}(k^2)$ vertices [31], both of which hold for general graphs. Fomin et al. [20] showed that Treewidth-t Vertex Deletion admits a kernel of size $k^{f(t)}$ in general graphs. Improving this result to a kernel of size $g(t) \cdot k^{\mathcal{O}(1)}$ has proven to be a significant challenge. If we manage to obtain a linear kernel on graphs of bounded expansion using a modulator to a bounded treewidth graph as parameter, then it would directly follow that Treewidth-t Vertex Deletion has a linear kernel in general graphs. This seems too good to be true.

Finally, many purely decision problems, such as Hamiltonian Path/Cycle and 3-Colorability, which have no natural parameter, are covered by our framework. It was already shown in [25] that these problems have a linear kernel on H-topological-minor-free graphs when parameterized by a modulator to bounded treewidth for H-topological minor free graphs. Taking a modulator to treedepth allowed us to extend this result to the class of bounded expansion graphs by choosing a modulator to bounded treedepth as a parameter.

7 Conclusions and Further Research

In this paper we presented kernelization results on graphs of bounded expansion, locally bounded expansion, and nowhere dense graphs. To the best of our knowledge, these are the very first kernelization results on these graph classes. The parameter that we use is the size of a modulator to constant treedepth graphs. Evidence suggests that any meta-theorem on linear kernels on graphs of bounded expansion that includes all the problems in Corollary 3 necessarily requires a parameter that cannot be weaker than what we have. However for problems whose solution sizes are not invariant under edge subdivisions, such as Dominating Set and Hamiltonian Cycle, it might be possible to obtain such a result.

There are some interesting open questions regarding the polynomial kernelizability of Longest Path. We conjecture that Longest Path has no polynomial kernel in general graphs with the size of a modulator to a single path (of arbitrary length) as parameter. This would show that if we use the size of a modulator to a (subgraph closed) graph property as parameter, then in general graphs there exists a dichotomy for Longest Path: If the graph property excludes long paths, there is a polynomial kernel; otherwise not. The polynomial kernel presented here has size $k^{g(d)}$, where k is the size of a treedepth-d modulator and $g(d) = 3^d$. Is there a kernel of size $g(d) \cdot k^{\mathcal{O}(1)}$, for some function g?

References

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

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

[3] H. L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25:1305–1317, 1996.

[4] H. L. Bodlaender. Treewidth: Algorithmic techniques and results. In *MFCS'97*, volume 1295 of *LNCS*, pages 19–36. Springer, 1997.

[5] H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *Journal of Computer and System Sciences*, 75(8):423–434, 2009.

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

[7] H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Kernel bounds for path and cycle problems. In *IPEC'11*, number 7112 in LNCS, pages 145–158. Springer, 2011.

[8] H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Kernel bounds for structural parameterizations of pathwidth. In *SWAT*, number 7357 in LNCS, pages 352–363. Springer, 2012.

[9] H. L. Bodlaender and T. Kloks. Better algorithms for the pathwidth and treewidth of graphs. In *ICALP'91*, volume 510 of *LNCS*, pages 544–555. Springer, 1991.

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

[11] J. Chen, I. A. Kanj, and W. Jia. Vertex cover: Further observations and further improvements. *Journal of Algorithms*, 41:280–301, 2001.

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

[13] R. Diestel. *Graph Theory.* Springer, Heidelberg, 4th edition, 2010.

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

[15] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer, 1999.

[16] Z. Dvořák and D. Král. Algorithms for classes of graphs with bounded expansion. In *WG'09*, volume 5911 of *LNCS*, pages 17–32. Springer, 2009.

[17] M. R. Fellows, B. M. P. Jansen, and F. Rosamond. Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity. *European J. Combin.*, 2012. In press.

[18] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer, 2006.

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

[20] F. V. Fomin, D. Lokshtanov, N. Misra, and S. Saurabh. Planar $\mathcal{F}$-Deletion: Approximation and Optimal FPT Algorithms. In *FOCS'12*, pages 470–479. IEEE Computer Society, 2012.

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

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

[23] B. M. P. Jansen and H. L. Bodlaender. Vertex cover kernelization revisited: Upper and lower bounds for a refined parameter. In *Proc. of 28th STACS*, volume 9 of *LIPIcs*, pages 177–188. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2011.

[24] B. M. P. Jansen and S. Kratsch. On polynomial kernels for structural parameterizations of odd cycle transversal. In *IPEC'11*, number 7112 in LNCS, pages 132–144. Springer, 2011.

[25] E. J. Kim, A. Langer, C. Paul, F. Reidl, P. Rossmanith, I. Sau, and S. Sikdar. Linear kernels and single-exponential algorithms via protrusion decomposition. Available at: `http://arxiv.org/abs/1207.0835`.

[26] J. Nešetřil and P. Ossona de Mendez. Grad and classes with bounded expansion I. Decompositions. *European J. Combin.*, 29(3):760–776, 2008.

[27] J. Nešetřil and P. Ossona de Mendez. First order properties on nowhere dense structures. *The Journal of Symbolic Logic*, 75(3):868–887, 2010.

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

[29] J. Nešetřil and P. Ossona de Mendez. *Sparsity: Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and Combinatorics.* Springer, 2012.

[30] J. Nešetřil, P. Ossona de Mendez, and D. R. Wood. Characterisations and examples of graph classes with bounded expansion. *Eur. J. Comb.*, 33(3):350–373, 2012.

[31] S. Thomassé. A quadratic kernel for feedback vertex set. In *SODA*, pages 115–119. SIAM, 2009.

[32] D. Wood. On the maximum number of cliques in a graph. *Graphs and Combinatorics*, 23:337–352, 2007.

8 Appendix

In this appendix, we define some of the problems that we mention in this paper.

Longest Path

Input: A graph G and a positive integer ℓ.

Problem: Does G contain a simple path of length at least ℓ?

Longest Cycle

Input: A graph G and a positive integer ℓ.

Problem: Does G contain a simple cycle of length at least ℓ?

Exact s, t-Path

Input: A graph G, two special vertices $s, t \in V(G)$ and a positive integer ℓ.

Problem: Is there a simple path in G from s to t of length exactly ℓ?

Exact Cycle

Input: A graph G and a positive integer ℓ.

Problem: Is there a simple cycle in G of length exactly ℓ?

Feedback Vertex Set

Input: A graph G and a positive integer ℓ.

Problem: Is there a vertex set $S \subseteq V(G)$ with at most ℓ vertices such that $G - S$ is a forest?

Treewidth

Input: A graph G and a positive integer ℓ.

Problem: Is the treewidth of G at most ℓ?

Pathwidth

Input: A graph G and a positive integer ℓ.

Problem: Is the pathwidth of G at most ℓ?

Treewidth-t Vertex Deletion

Input: A graph G and a positive integer ℓ.

Problem: Is there a vertex set $S \subseteq V(G)$ with at most ℓ vertices such that the treewidth of $G - S$ is at most t?

Dominating Set

Input: A graph $G = (V, E)$ and a positive integer ℓ.

Problem: Is there a vertex set $S \subseteq V$ with at most ℓ vertices such that for all $u \in V \setminus S$ there exists $v \in S$ such that $uv \in E$?

If in addition, we require that $G[S]$ is a connected graph then the problem is called Connected Dominating Set.

r-Dominating Set

Input: A graph $G = (V, E)$ and a positive integer ℓ.

Problem: Is there a vertex set $S \subseteq V$ with at most ℓ vertices such that for all $u \in V \setminus S$ there exists $v \in S$ such that $d(u, v) \leqslant r$?

Efficient Dominating Set

Input: A graph $G = (V, E)$ and a positive integer ℓ.

Problem: Is there an independent set $S \subseteq V$ with at most ℓ vertices such that for every $u \in V \setminus S$ there exists exactly one $v \in S$ such that $uv \in E$?

Edge Dominating Set

Input: A graph $G = (V, E)$ and a positive integer ℓ.

Problem: Is there an edge set $S \subseteq E$ of size at most ℓ such that for every $e \in E \setminus S$ there exists $e' \in S$ such that e and e' share an endpoint?

Induced Matching

Input: A graph $G = (V, E)$ and a positive integer ℓ.

Problem: Is there an edge set $S \subseteq E$ of size at least ℓ such that S is a matching and for all $u, v \in V(S)$, if $uv \in E$ then $uv \in S$?

Chordal Vertex Deletion

Input: A graph $G = (V, E)$ and a positive integer ℓ.

Problem: Is there a vertex set $S \subseteq V$ of size at most ℓ such that $G - S$ is chordal?