Boundaried Kernelization

Leonid Antipov
Humboldt-Universität zu Berlin, Berlin, Germany
leonid.antipov@hu-berlin.de

Stefan Kratsch
Humboldt-Universität zu Berlin, Berlin, Germany
kratsch@informatik.hu-berlin.de

April 28, 2025

Abstract

The notion of a (polynomial) kernelization from parameterized complexity is a well-studied model for efficient preprocessing for hard computational problems. By now, it is quite well understood which parameterized problems do or (conditionally) do not admit a polynomial kernelization. Unfortunately, polynomial kernelizations seem to require strong restrictions on the *global structure* of inputs.

To avoid this restriction, we propose a model for efficient *local preprocessing* that is aimed at local structure in inputs. Our notion, dubbed *boundaried kernelization*, is inspired by protrusions and protrusion replacement, which are tools in meta-kernelization [Bodlaender et al. J'ACM 2016]. Unlike previous work, we study the preprocessing of suitable boundaried graphs in their own right, in significantly more general settings, and aiming for polynomial rather than exponential bounds. We establish polynomial boundaried kernelizations for a number of problems, while *unconditionally* ruling out such results for others. We also show that boundaried kernelization can be a tool for regular kernelization by using it to obtain an improved kernelization for VERTEX COVER parameterized by the vertex-deletion distance to a graph of bounded treedepth.

1 Introduction

We introduce *boundaried kernelization* as a model for provably efficient *local preprocessing*, i.e., preprocessing that leverages local structure without having to explicitly consider the entire instance. This extends the notion of *kernelization* from *parameterized complexity* and is inspired by *protrusions* and *protrusion replacement*, which have been used, in particular, in meta theorems for kernelization (e.g., [6]). Unlike protrusions, which are subgraphs of constant treewidth that have a constant-size interface/boundary to the rest of the graph, we study different kinds of internal structure and we allow for larger boundaries. Further, we usually aim to reduce the size to at most polynomial in boundary size and optional further parameters, while for protrusion replacement it is sufficient to reduce to a size with any (usually at least exponential) dependence on treewidth and boundary size. Let us first give some context.

Parameterized complexity studies *parameterized problems*, which are classical problems augmented with one or more parameters such as solution size, dimension, or treewidth; often, parameters quantify structural properties. A central goal is to design algorithms that are fast when the chosen parameter value(s) are low, showing that the corresponding property or structure is algorithmically beneficial (called *fixed-parameter tractable* algorithms). This parameterized setting also allows for a robust notion of efficient preprocessing: A *kernelization* for a parameterized problem is an efficient algorithm that expects an instance (x, k), with parameter k, and returns an equivalent instance (x', k') of size upper bounded by some computable function $f(k)$. In particular, one is interested what parameterized problems admit *polynomial kernelizations* where the function f is polynomially bounded. Kernelization has been intensely studied (see, e.g., recent works [4,

5, 11, 19, 24, 25, 36, 38, 39, 43, 44, 47] or the dedicated book [28]) and, by now, it is quite well understood which problems do or (conditionally) do not admit polynomial kernelizations. Moreover, there are both meta results (e.g., [6, 27, 29, 45]), giving upper bounds for whole classes of problems, as well as dichotomy results (e.g., [41, 40, 18, 11]), classifying all problems in some class into admitting or (conditionally) not admitting a polynomial kernelization.

We see *two main motivations* for studying local preprocessing: *First*, kernelizations usually rely on *reduction rules*, which usually apply to local configurations in the input. For example, this includes simple rules like *folding of degree-two vertices* for VERTEX COVER [14] and seeking *flowers*, i.e., cycles that pairwise intersect in the same vertex, for FEEDBACK VERTEX SET [10, 46]. In other words, depending on the kind of reduction rule, the scope of the rule may vary, but beyond the required scope it usually does not matter what else is present in the instance.[1] Is there a sensible way of capturing local structure and giving local guarantees for the effectiveness of preprocessing? *Second*, most polynomial kernelizations in the literature are for parameterization by solution size or by the distance to some tractable special case for the problem. E.g., we are given a graph G and a modulator $X \subseteq V(G)$ such that $G - X$ belongs to a graph class where the target problem can be solved efficiently, with $|X|$ being the parameter. Intuitively, unless $\mathsf{P} = \mathsf{NP}$, efficient preprocessing can at best deal with tractable parts so the parameter must directly or indirectly allow to (at least) bound the size of the remainder of the instance, which may be structured arbitrarily. Together, this yields a fairly strict requirement for kernelization to be useful: Almost the entire instance must match some tractable special case for the problem in question. What if only parts of an instance fulfill such a restriction? Do we still get nontrivial guarantees for efficient preprocessing?

Our work. Inspired by protrusions and protrusion replacement from meta kernelization [6], we use *boundaried graphs* to define *boundaried kernelization* (see Section 3 for formal definitions): A boundaried graph G_B is a graph G with the additional specification of a set $B \subseteq V(G)$ of boundary vertices. Informally, a boundaried kernelization for, e.g., the problem VERTEX COVER[fvs], i.e., VERTEX COVER parameterized by the size of a given feedback vertex set, is a polynomial-time algorithm that expects as input a boundaried graph G_B with forest-modulator X of size k, and returns a boundaried graph G'_B of size bounded by a function of parameter k and boundary size $|B|$ such that attaching the same graph H_B to either G_B or G'_B (by identification of boundary vertices, also called *gluing*) will yield graphs that are equivalent with respect to VERTEX COVER. In this way, local preprocessing for VERTEX COVER with respect to local structure as quantified by modulator X corresponds to a boundaried kernelization for VERTEX COVER[fvs]. Note that for $B = \emptyset$ and gluing the empty graph to G_B this includes the requirement that G and G' are equivalent for VERTEX COVER (and G' is of size bounded by a function of k), which essentially constitutes a regular kernelization for VERTEX COVER[fvs] (cf. Lemma 4). Accordingly, we mainly work with problems that admit a polynomial kernelization.

We establish polynomial size boundaried kernelizations for several problems that are known to admit regular polynomial kernelizations. This confirms the intuition that the local nature of reduction rules often allows us to work on the well-structured part only and ignore the rest of the instance.

Theorem 1. *The following parameterized problems admit a polynomial boundaried kernelization:* VERTEX COVER[vc]*,* VERTEX COVER[fvs]*,* FEEDBACK VERTEX SET[fvs]*,* LONG CYCLE[vc]*,* LONG PATH[vc]*, and* HAMILTONIAN CYCLE[vc]*,* HAMILTONIAN PATH[vc] *as well as* HAMILTONIAN CYCLE/PATH$[\#v, \deg(v) \neq 2]$.

We complement this by proving *unconditionally* that some problems do not admit polynomial boundaried kernelizations despite admitting a regular polynomial kernelization. To round out the picture we also prove unconditionally that DOMINATING SET[vc], i.e., DOMINATING SET parameterized by the size of a given vertex cover, admits no polynomial boundaried kernelization, while it was already known to not admit a polynomial kernelization unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

[1] A notably different case: Kernelizations obtained via cut-covering sets and other matroid-based techniques [42] use the entire instance to discover safe reduction steps.

Theorem 2. *The following parameterized problems do not admit a polynomial boundaried kernelization:* CLUSTER EDITING[cvd], CLUSTER EDITING[ce], MAXIMUM CUT[vc], TREE DELETION SET[vc], TREE DELETION SET[tds], LONG CYCLE[$\#v, \deg(v) \neq 2$], LONG PATH[$\#v, \deg(v) \neq 2$], DOMINATING SET[vc].

Finally, we show that, like protrusion replacement, boundaried kernelization can itself be useful for obtaining regular kernelization results. Concretely, we improve the size of a kernelization for VERTEX COVER[mod to td $\leq d$], i.e., VERTEX COVER parameterized by the size of a given deletion set/modulator to treedepth at most d, from $\mathcal{O}(k^{2^{\Theta(d^2)}})$ [12, 32] to $\mathcal{O}(k^{2^{d-1}})$ vertices, which is a significant improvement and comes much closer to the known lower bound of $\mathcal{O}(k^{2^{d-2}+1})$ vertices [32].

Theorem 3. *For any constant $d \in \mathbb{N}_{\geq 1}$, the parameterized problem* VERTEX COVER[mod to td $\leq d$] *admits a polynomial kernelization with $\mathcal{O}(k^{2^{d-1}})$ vertices.*

Related work. Arnborg et al. [2] used finite index of gluing-equivalence (defined in Section 3) and protrusion replacement for the construction of algorithms that solve the membership problem for MSO-definable graph classes with bounded treewidth. Likewise, Fellows and Langston [22] showed that under condition of finite index of gluing-equivalence, the obstruction set of a minor-closed graph class can be computed as soon as a treewidth bound for the former is known. Bodlaender and van Antwerpen-de Fluiter [9] extended the use to several decision, construction, and optimization problems on graphs of bounded treewidth.

As a tool for obtaining linear kernelizations for (CONNECTED) DOMINATING SET on graphs with a fixed excluded topological minor, Fomin et al. [26] introduce a notion dubbed "generalized protrusion:" This refers to a subgraph with constant-size boundary but, instead of having constant treewidth, it contains at most a constant number of vertices from the sought solution (presently, from the sought minimum (connected) dominating set). In a work on kernelization for INTEGER LINEAR PROGRAM FEASIBILITY, Jansen and Kratsch [35] consider shrinking protrusion-like subsystems of ILPs with a constant-size boundary that are either totally unimodular or have a Gaifman graph of bounded treewidth.

Other prior work has considered "dynamic sketching" [3], respectively "preprocessing under uncertainty" [21, 20], mostly for tractable problems, where parts of the input are dynamic or unknown and the preprocessing needs to be consistent with any possible instantiation of the dynamic/unknown part, e.g., adjacency of vertices in dynamic part may change. Here the dynamic/unknown part is similar in spirit to the concept of a boundary, though the effects that either can have on the overall solution are very different.

Organization. Section 2 contains preliminaries regarding (graph) problems. In Section 3 we define boundaried graphs, introduce boundaried kernelization and obtain some first results. Several polynomial boundaried kernelization results are given in Section 4, while in Section 5 we unconditionally rule out existence of (polynomial) boundaried kernelization for several problems. In Section 6 we apply the notion of boundaried kernelization in order to show an improved polynomial kernelization for VERTEX COVER[mod to td $\leq d$]. We conclude in Section 7.

2 Preliminaries

Decision and optimization problems. A *decision problem* is a set $\Pi \subseteq \Sigma^*$. We call any $x \in \Sigma^*$ an *instance* of Π. An *optimization problem* is a function $\Pi\colon \Sigma^* \times \Sigma^* \to \mathbb{N} \cup \{\pm\infty\}$. Again, an instance of Π is defined as any $x \in \Sigma^*$. For some instance x and some $s \in \Sigma^*$, if $\Pi(x, s) \neq \pm\infty$, i.e., $\Pi(x, s) \in \mathbb{N}$, then s is called a *(feasible) solution* for Π on x, and $\Pi(x, s)$ is called the *value* of s. Optimization problems are divided into *minimization* and *maximization* problems. For a minimization problem we define the *optimum value* for Π on instance x, denoted $\mathrm{OPT}_\Pi(x)$, or simply $\mathrm{OPT}(x)$ if Π is clear from context, as the lowest value of a feasible solution s on x. Conversely, for a maximization problem we define $\mathrm{OPT}_\Pi(x)$ as the highest value of a feasible solution on x. If there is no feasible solution for Π on x, then $\mathrm{OPT}_\Pi(x) = +\infty$ for a mininimization problem, resp. $\mathrm{OPT}_\Pi(x) = -\infty$ for a maximization problem. A solution with optimum value is called an *optimum solution*. For optimization problems used throughout this work, the *value* $\Pi(x, s)$ of solution s on

instance x is simply the *size* of s (denoted $|s|$). We remark, however, that in general this does not need to be the case. We call an optimization problem Π an NP-*optimization problem*, if there exists some polynomial function p, such that for every instance $x \in \Sigma^*$ the following are upper bounded by $p(|x|)$: (i) the size of any feasible solution $s \in \Sigma^*$ on x; (ii) the time needed to compute $\Pi(x, s)$ for any feasible s; (iii) the time needed to check if $s \in \Sigma^*$ with $|s| \leq p(|x|)$ is feasible. Note that by (ii), for such Π the value of a feasible solution s for x can be encoded in length at most $p(|x|)$. For any optimization problem Π we define the decision problem Π_d such that for any instance $(x, k) \in \Sigma^* \times \mathbb{N}$ it holds that $(x, k) \in \Pi_d$ if and only if there exists a feasible solution with value at most k for minimization problems, resp., at least k for maximization problems.

Parameterized complexity. A *parameterized problem* is a set $Q \subseteq \Sigma^* \times \mathbb{N}$. For a tuple $(x, k) \in \Sigma^* \times \mathbb{N}$, k is called the *parameter*. For an optimization problem Π, its *standard parameterization* is the decision variant Π_d with sought solution value k as the parameter, denoted by $\Pi[k]$. In contrast to this, for Π being some decision/optimization problem and ρ some minimization problem, we define the *structurally parameterized problem* $\Pi[\rho]$, for which it holds that $(x, s, k) \in \Pi[\rho]$ if and only if s is a feasible solution of value at most k for ρ on x, and (i) for Π being a decision problem, $x \in \Pi$; (ii) for Π being an optimization problem, $x \in \Pi_d$, i.e., $x = (x', \ell)$ where ℓ is the sought solution value for Π on x'.

Let Q be a parameterized problem and let f be a computable function. A *kernelization* for Q of *size* f is a polynomial-time algorithm which, given as input (x, k), outputs (x', k') such that $|x|, k' \leq f(k)$ and $(x, k) \in Q \Leftrightarrow (x', k') \in Q$. If f is a polynomial, the algorithm is called a *polynomial kernelization*. We will use the term *kernel* interchangeably with the term *kernelization*.

Graphs and graph problems. We use standard graph theoretic notation mostly following Diestel [16]. We denote the classes of independent and complete graphs, trees and forests by $\mathcal{C}^{\texttt{independent}}, \mathcal{C}^{\texttt{complete}}, \mathcal{C}^{\texttt{tree}}$ and $\mathcal{C}^{\texttt{forest}}$, respectively. Any graph class closed under taking subgraphs is said to be *hereditary*. A *graph problem* is a problem for which every instance $x = (G, \hat{x})$ consists of a graph G and the remaining string $\hat{x}$. We say that Π is a *pure graph problem* if the remaining string $\hat{x}$ is empty, i.e., every instance is simply a graph G, and Π is isomorphism-invariant. For some graph class $\mathcal{C}$ and graph G, a $\mathcal{C}$-modulator (or vertex deletion set to $\mathcal{C}$) for G is a vertex set $X \subseteq V(G)$ such that $G - X \in \mathcal{C}$. We denote the problem of finding a minimum size modulator to $\mathcal{C}$ by MOD TO $\mathcal{C}$. All other problems discussed in this work are defined in Appendix A.

The *treedepth* of a graph G, denoted $\mathrm{td}(G)$, is the minimum height of a rooted forest F, called a *treedepth decomposition* of G, such that if $\{u, v\}$ is an edge in G, then u is an ancestor of v or vice versa. There is also a recursive definition for the treedepth of G, which basically constructs the treedepth decomposition F of G: (i) if G is an independent set, then $\mathrm{td}(G) = 1$; (ii) else if G is disconnected, then $\mathrm{td}(G) = \max\{\mathrm{td}(C) \mid C \text{ is a component of } G\}$; (iii) else $\mathrm{td}(G) = \min\{\mathrm{td}(G - v) + 1 \mid v \in V(G)\}$. Similar to our other graph class notation, we denote by $\mathcal{C}^{\mathrm{td} \leq d}$ the class of graphs whose treedepth is at most d.

3 Boundaried kernelization

3.1 Boundaried graphs and related terms

The notion of boundaried graphs and many of the related definitions were already given, e.g., by Bodlaender and van Antwerpen-de Fluiter [9], and extensively used for the notion of protrusions and the related meta kernelization [6]. We state our definitions on this topic and follow them by two technical but simple lemmas.

A *(B-)boundaried graph* consists of a graph G and a *boundary* $B \subseteq V(G)$, and is denoted as G_B. Two boundaried graphs G_B, G'_B are *isomorphic*, if there exists an isomorphism from G to G' that identifies vertices in B. A boundaried graph G'_B is said to be a subgraph of G_B, if G' is a subgraph of G.

For graph class $\mathcal{C}$ we denote by $\mathcal{C}_B$ the class of all $\mathcal{C}$-modulated B-boundaried graphs, i.e., all boundaried graphs G_B for which it holds that $G - B \in \mathcal{C}$.

Given two boundaried graphs G_B and H_C, the *gluing* operation on those boundaried graphs, denoted as $G_B \oplus H_C$, results in a new (simple) graph, which is the disjoint union of G and H while identifying vertices from $B \cap C$. In particular, if an edge e between vertices in $B \cap C$ is contained in both G and H, the resulting graph contains e only once. For convenience, we will tacitly assume that $V(G) \cap V(H) \subseteq B \cap C$, unless mentioned otherwise. A few examples of graph gluing are given in Figure 1. We remark that for boundaried graphs F_A, G_B, H_C clearly $(F_A \oplus G_B)_{A \cup B} \oplus H_C$ is isomorphic to $F_A \oplus (G_B \oplus H_C)_{B \cup C}$ and that $F_A \oplus G_B$ is isomorphic to $G_B \oplus F_A$, i.e., in this sense graph gluing is associative and commutative. It also preserves isomorphism, i.e., if G_B is isomorphic to G'_B and H_C to H'_C, then also $G_B \oplus H_C$ is isomorphic to $G'_B \oplus H'_C$.

Let Π be a pure graph problem, let G_B and G'_B be boundaried graphs. We say that G_B and G'_B are *gluing equivalent with respect to* Π *and* B, denoted $G_B \equiv_{\Pi,B} G'_B$, if: (i) for Π being a decision problem, it holds for every H_B, that $G_B \oplus H_B \in \Pi \Leftrightarrow G'_B \oplus H_B \in \Pi$; (ii) for Π being an optimization problem, there exists some constant $\Delta \in \mathbb{Z}$ such that for every H_B it holds that $\mathrm{OPT}_\Pi(G_B \oplus H_B) = \mathrm{OPT}_\Pi(G'_B \oplus H_B) + \Delta$. In the latter case we also say that G_B and G'_B are gluing equivalent with *offset* $\pm\Delta$, tacitly having fixed one direction. For some graph class $\mathcal{C}$ the equivalence relation $\equiv^{\mathcal{C}}_{\Pi,B}$ (resp. $\equiv^{\mathcal{C}_B}_{\Pi,B}$) is the restriction of $\equiv_{\Pi,B}$ to $\mathcal{C}$ (resp. $\mathcal{C}_B$). If Π, B, and $\mathcal{C}$ or $\mathcal{C}_B$ are clear from context, we write $\equiv$.

Some pure graph decision problem Π has *finite index*, if there exists some function $f\colon \mathbb{N} \to \mathbb{N}$, such that for each boundary B the number of equivalence classes of $\equiv_{\Pi,B}$ is at most $f(|B|)$. In the case of Π being an optimization problem, Π is said to have *finite integer index*. For graph class $\mathcal{C}$ we say that Π has *finite (integer) index on* $\mathcal{C}$ (resp. on $\mathcal{C}_B$), if there exists some function $f\colon \mathbb{N} \to \mathbb{N}$ such that for each boundary B the number of equivalence classes of $\equiv^{\mathcal{C}}_{\Pi,B}$ (resp. $\equiv^{\mathcal{C}_B}_{\Pi,B}$) is at most $f(|B|)$. Among problems with finite (integer) index, we highlight those having *single-exponential (integer) index*, which in addition requires some constant c and polynomially bounded function f such that $\equiv_{\Pi,B}$ (resp. $\equiv^{\mathcal{C}}_{\Pi,B}$ and $\equiv^{\mathcal{C}_B}_{\Pi,B}$) has $\mathcal{O}(c^{f(|B|)})$ equivalence classes.

The following lemma allows us to use gluing-equivalence for an artificially increased boundary in order to obtain automatically also gluing-equivalence for the actually needed boundary. After that, we state Lemma 2 which will be used in Section 6. Note that problem Π can be either a decision or an optimization problem. For the proof of Lemma 1 we write the proof for both cases. However, since there is not much additional argumentation needed for the case of a decision problem, in subsequent proofs we will assume that Π is an optimization problem.

Lemma 1. *Let Π be a pure graph problem, G and G' graphs, and B, C vertex subsets of both $V(G)$ and $V(G')$ such that $B \subseteq C$. Then $G_C \equiv_{\Pi,C} G'_C$ implies $G_B \equiv_{\Pi,B} G'_B$, with the same offset Δ for these two equivalences, if Π is an optimization problem.*

Proof. Assume first that Π is an optimization problem and that the stated condition holds, i.e., for some constant Δ and every H_C it holds that $\mathrm{OPT}_\Pi(G_C \oplus H_C) = \mathrm{OPT}_\Pi(G'_C \oplus H_C) + \Delta$. Now to show that for any H_B it also holds that $\mathrm{OPT}_\Pi(G_B \oplus H_B) = \mathrm{OPT}_\Pi(G'_B \oplus H_B) + \Delta$, fix an arbitrary B-boundaried graph H_B. Since Π is isomorphism invariant, we assume without loss of generality, that the vertices in common between H and G, resp., H and G', are all contained in B. As a result, also no vertices in $C \setminus B$ are in common between H and G, G'. From H we construct the graph H' by adding the vertices in $C \setminus B$ as an independent set, which leads to the fact that $G_B \oplus H_B = G_C \oplus H'_C$ and $G'_B \oplus H_B = G'_C \oplus H'_C$. This way we obtain the needed equivalence $\mathrm{OPT}_\Pi(G_B \oplus H_B) = \mathrm{OPT}_\Pi(G_C \oplus H'_C) = \mathrm{OPT}_\Pi(G'_C \oplus H'_C) + \Delta = \mathrm{OPT}_\Pi(G'_B \oplus H_B) + \Delta$. Now assume that Π is a decision problem. Again, for any H_B it holds that $G_B \oplus H_B = G_C \oplus H'_C$ and $G'_B \oplus H_B = G'_C \oplus H'_C$, which makes $G_B \oplus H_B \in \Pi \Leftrightarrow G'_B \oplus H_B \in \Pi$ follow from the fact that $G_C \oplus H'_C \in \Pi \Leftrightarrow G'_C \oplus H'_C \in \Pi$. □

Lemma 2. *Let Π be a pure graph problem, B some vertex set, and G^1, G^2, $\hat{G}^1$, $\hat{G}^2$ graphs, such that $G^1_B \equiv_{\Pi,B} \hat{G}^1_B$ and $G^2_B \equiv_{\Pi,B} \hat{G}^2_B$. Then it holds that $(G^1_B \oplus G^2_B)_B \equiv_{\Pi,B} (\hat{G}^1_B \oplus \hat{G}^2_B)_B$.*
If Π is an optimization problem and Δ_1, Δ_2 are the offset values for the equivalences between G^1_B and $\hat{G}^1_B$, resp. G^2_B and $\hat{G}^2_B$, when glued to any H_B, then the offset for the equivalence between $(G^1_B \oplus G^2_B)_B$ and $(\hat{G}^1_B \oplus \hat{G}^2_B)_B$ equals $\Delta_1 + \Delta_2$.

Proof. Without loss of generality assume that Π is an optimization problem. Let H_B be any B-boundaried graph. From $G^1_B \equiv_{\Pi,B} \hat{G}^1_B$ follows existence of some Δ_1 such that $\mathrm{OPT}(G^1_B \oplus (G^2_B \oplus H_B)_B) = \mathrm{OPT}(\hat{G}^1_B \oplus (G^2_B \oplus H_B)_B) + \Delta_1$. At the same time $G^2_B \equiv_{\Pi,B} \hat{G}^2_B$ implies some Δ_2 such that $\mathrm{OPT}(G^2_B \oplus (\hat{G}^1_B \oplus H_B)_B) = \mathrm{OPT}(\hat{G}^2_B \oplus (\hat{G}^1_B \oplus H_B)_B) + \Delta_2$. Together with commutativity and associativity of graph gluing, this yields $\mathrm{OPT}((G^1_B \oplus G^2_B)_B \oplus H_B)) = \mathrm{OPT}((\hat{G}^1_B \oplus G^2_B)_B \oplus H_B) + \Delta_1 = \mathrm{OPT}((\hat{G}^1_B \oplus \hat{G}^2_B)_B \oplus H_B) + \Delta_1 + \Delta_2$. □

3.2 Boundaried kernelization

We will now formally introduce our notion of boundaried kernelization. Presently, we do this for the restricted setting of pure graph problems Π parameterized by some pure graph minimization problem ρ. This enables us to prove a few general results on boundaried kernelizations and captures most structural parameterizations while also keeping the required notation somewhat concise. Possible extensions of the notion are briefly discussed in the conclusion. We remark that while we work with pure graph problems, the boundaried kernelization gets as input a *boundaried* graph G_B. This is consistent since the result of gluing G_B with any other boundaried graph H_B results in the *simple* graph $Q = G_B \oplus H_B$, which is then the actual input to the pure graph problem, for which the preprocessing was done on the known local part G_B.

Unlike in regular kernelization, where, for Π being an optimization problem, a sought solution size k is given and can be changed by the kernel (adjusting to situations like safeness of including some vertex in the solution), this is not the case for boundaried kernelization, where only local structure of G_B is used, independently of what graph H_B could be glued to G_B and what the global solution size could be. For this reason, the boundaried kernelization needs to output its influence on the solution size in form of the solution size offset Δ, which does not need to be bounded. Note that this Δ is already part of the definition of gluing equivalence.

Definition 1 (Boundaried kernelization)**.** *Let Π be a pure graph problem, ρ a pure graph minimization problem, and f a computable function. A boundaried kernelization of size f for $\Pi[\rho]$ is a polynomial-time algorithm that, given a boundaried graph G_B and ρ-solution s, outputs a $\equiv_{\Pi,B}$-equivalent boundaried graph G'_B with ρ-solution s', such that $|G'|, |s'|, \rho(G', s') \leq f(|B| + \rho(G, s))$. Additionally, if Π is an optimization problem, the algorithm also needs to output an integer Δ, such that for all H_B it holds that $\mathrm{OPT}_\Pi(G_B \oplus H_B) = \mathrm{OPT}_\Pi(G'_B \oplus H_B) + \Delta$. If f is upper bounded by some polynomial function, the boundaried kernelization has polynomial size and is called a polynomial boundaried kernelization.*

Later we will show polynomial boundaried kernelizations for several problems parameterized by the size of a given modulator $|X|$ to some hereditary graph class $\mathcal{C}$. As the size of the output should be bounded polynomially in $|B| + |X|$, it suffices to bound/reduce the size of $R := V(G) \setminus (B \cup X)$. In order to simplify this, we show that it suffices to deal with the special case where B itself is a modulator to $\mathcal{C}$, i.e., where $G - B \in \mathcal{C}$.

Lemma 3. *Let Π be a pure graph problem, $\mathcal{C}$ be a hereditary graph class and f a computable function. If there exists a boundaried kernelization of size $f(|B|)$ for Π[mod to $\mathcal{C}$] with the additional restriction that B coincides with the given and output solutions to* MOD TO $\mathcal{C}$*, then there also exists a boundaried kernelization of size f for Π[mod to $\mathcal{C}$] without that restriction.*

Proof. Without loss of generality assume that Π is an optimization problem. Let G_B be the boundaried graph and X the $\mathcal{C}$-modulator, which are given to the unrestricted boundaried kernelization. Since $\mathcal{C}$ is hereditary per requirement, it also holds that $B \cup X$ is a $\mathcal{C}$-modulator. Thus we can apply the restricted boundaried kernelization on boundaried graph $G_{B\cup X}$ and $\mathcal{C}$-modulator $B \cup X$, which outputs a boundaried graph $G'_{B\cup X}$, the $\mathcal{C}$-modulator $B \cup X$, and an integer Δ, such that $|G'|, |B \cup X| \leq f(|B \cup X|)$ and for all $H_{B\cup X}$ it holds that $\mathrm{OPT}_\Pi(G_{B\cup X} \oplus H_{B\cup X}) = \mathrm{OPT}_\Pi(G'_{B\cup X} \oplus H_{B\cup X}) + \Delta$. By Lemma 1 it becomes clear that for the same Δ it also holds for every H_B that $\mathrm{OPT}_\Pi(G_B \oplus H_B) = \mathrm{OPT}_\Pi(G'_B \oplus H_B) + \Delta$. Therefore it suffices for the unrestricted boundaried kernelization to output the boundaried graph G'_B and its $\mathcal{C}$-modulator $B \cup X$. □

The following lemma shows that, for parameterization by NP-minimization problem ρ, a boundaried kernelization of size f also implies a (regular) kernelization of size at most polynomial in f. In this sense, boundaried kernelizations are stronger than (regular) kernelizations, showing, in particular, that a polynomial kernelization is a prerequisite for admitting a polynomial boundaried kernelization. We note that most structural parameterizations are indeed based on NP-minimization problems, as one wishes to be able verify the (often provided) structure efficiently. (Parameterization by maximization problems is often unwieldy to deal with, e.g., because we need optimum solutions to get structural implications, but optimality is usually hard to verify. Nevertheless, parameterization by max leaf number comes to mind as a useful exception, cf. [23].)

Lemma 4. *Let Π and ρ be pure graph problems. Additionally, let ρ be an NP-minimization problem, and Π either a decision problem or an NP-optimization problem. If $\Pi[\rho]$ admits a (polynomial) boundaried kernelization, it also admits a (polynomial) kernelization.*

Proof. First, we prove the lemma for Π being an NP-optimization problem. We are given an instance (G, ℓ, s, k), i.e., s is a solution with value at most k for ρ on G and the question is if Π has a solution for G with value at most ℓ if Π is a minimization problem, resp. with value at least ℓ if Π is a maximization problem. Let us first make sure that the encoding of ℓ is not overly large: By the definition of NP-optimization problems, there exists some polynomial function p for Π, such that any feasible solution for Π on G has size at most $p(|G|)$ and its value can be encoded with length at most $p(|G|)$. Hence, if ℓ is too large for such an encoding, and if Π is a minimization problem, then there exists a feasible solution with value at most ℓ if and only if there exists some solution with a value that can be encoded with length $p(|G|)$, so we can work further with the instance (G, ℓ', s, k) instead, where ℓ' is the highest possible number that can be encoded with length $p(|G|)$; and if Π is a maximization problem, then there cannot be a feasible solution with value at least ℓ, so we can output some fixed NO-instance of constant size. So from now it holds for our instance (G, ℓ, s, k) that ℓ can be encoded with length at most $p(|G|)$.

For any boundary $B \subseteq V(G)$, gluing G_B to the independent set on vertex set B results in G itself. In particular this also works for B being empty. We apply the boundaried kernelization of size f on $G_\emptyset$ and s, and get as output a boundaried graph $G'_\emptyset$, a ρ-solution s', and an integer Δ such that for all $H_\emptyset$ it holds that $\mathrm{OPT}_\Pi(G_\emptyset \oplus H_\emptyset) = \mathrm{OPT}_\Pi(G'_\emptyset \oplus H_\emptyset) + \Delta$. By choosing $H_\emptyset$ as the empty graph, it holds in particular that $\mathrm{OPT}_\Pi(G) = \mathrm{OPT}_\Pi(G') + \Delta$. Both $G'_\emptyset$ and s' have size and value at most $f(0 + \rho(G, s)) \leq f(k)$. If $\Delta > \ell$ and Π is a minimization problem, we immediately recognize that there cannot be a feasible solution with value at most $\ell - \Delta < 0$ for G' and thus, by gluing equivalence, no feasible solution with value at most ℓ exists for G. Conversely, if $\Delta > \ell$ and Π is a maximization problem, then there exists a feasible solution with value at least $\ell - \Delta$ if and only if there exists one with value at least 0; so we can replace Δ by ℓ. Similar to the first paragraph, we can now make sure that $\ell - \Delta$ can be encoded with length at most $p(|G'|) \leq p(f(k))$, and otherwise replace $\ell - \Delta$ by the highest possible number that can be encoded with length $p(|G'|)$ or output a fixed NO-instance, depending on whether Π is a minimization or maximization problem. Altogether, the instance $(G', \ell - \Delta, s', \rho(G', s'))$ is a correct output for a (polynomial) kernelization, as it satisfies the following: (i) $(G, \ell, s, k) \in \Pi[\rho] \Leftrightarrow (G', \ell - \Delta, s', \rho(G', s')) \in \Pi[\rho]$; and (ii) $|G'|, |s'|, \rho(G', s')$ and the encoding length of $|\ell - \Delta|$ are all upper bounded by $f(k) + p(f(k))$.

Now let Π be a decision problem. We are given an instance (G, s, k) this time, with s a solution with value at most k for ρ on G and the question being if $G \in \Pi$. Similar to above, we apply the boundaried kernelization of size f on $G_\emptyset$ and s, in order to obtain a boundaried graph $G'_\emptyset$ and its ρ-solution s', such that for all $H_\emptyset$ it holds that $G_\emptyset \oplus H_\emptyset \in \Pi \Leftrightarrow G'_\emptyset \oplus H_\emptyset \in \Pi$. Again, by choosing $H_\emptyset$ as the empty graph, it holds in particular that $G \in \Pi \Leftrightarrow G' \in \Pi$. Both G' and s' have size and value at most $f(0 + \rho(G, s)) \leq f(k)$. Altogether, the instance $(G', s', \rho(G', s'))$ is a correct output for a (polynomial) kernelization, as it satisfies the following: (i) $(G, s, k) \in \Pi[\rho] \Leftrightarrow (G', s', \rho(G', s')) \in \Pi[\rho]$; and (ii) $|G'|, |s'|$ and $\rho(G', s')$ are all upper bounded by $f(k)$. □

The next and final lemma of this section will be used to rule out (polynomial) boundaried kernelizations. In contrast to lower bounds for (regular) kernelization (cf. [8, 15]), working with the index of the corresponding equivalence classes allows us to get unconditional lower bounds. In particular, it follows that if

Π[mod to $\mathcal{C}$] has a polynomial boundaried kernelization, then $\equiv^{\mathcal{C}_B}_{\Pi,B}$ and $\equiv^{\mathcal{C}}_{\Pi,B}$ have single-exponential index (and, conversely, larger index rules out such a boundaried kernelization).

Lemma 5. *Let Π be a pure graph problem, ρ a pure graph minimization problem and f a computable function such that $\Pi[\rho]$ admits a boundaried kernelization of size f. Further fix some vertex set B and graph class $\mathcal{C}$ such that for each $G \in \mathcal{C}$ with $B \subseteq V(G)$ exists some ρ-solution s with $\rho(G,s) \leq g(|B|)$ for some function g. Then the number of equivalence classes of $\equiv^{\mathcal{C}}_{\Pi,B}$ is upper bounded by $\mathcal{O}(2^{f(|B|+g(|B|))^2})$.*

Proof. Let F_B, G_B be two boundaried graphs with $F, G \in \mathcal{C}$ that are not equivalent with respect to $\equiv^{\mathcal{C}}_{\Pi,B}$. Let s_F, s_G be respective ρ-solutions with value at most $g(|B|)$, which exist by assumption of the lemma. Then the given boundaried kernelization outputs for F_B, s_F and G_B, s_G two graphs F'_B and G'_B which are not isomorphic, as otherwise it would hold that $F_B \equiv_{\Pi,B} F'_B \equiv_{\Pi,B} G'_B \equiv_{\Pi,B} G_B$, leading to a contradiction. The number of equivalence classes of $\equiv^{\mathcal{C}}_{\Pi,B}$ is thus upper bounded by the number of non-isomorphic graphs we might get as output from the boundaried kernelization when given some B-boundaried graph in $\mathcal{C}$ and respective solution s with guaranteed value at most $g(|B|)$. Since these graphs have size (and thus number of vertices) at most $q = f(|B| + g(|B|))$, their number is upper bounded by $\sum_{n=0}^{q} \sum_{m=0}^{\binom{n}{2}} \binom{\binom{n}{2}}{m} \leq q \cdot 2^{\binom{q}{2}} \in \mathcal{O}(2^{q^2})$. □

4 Upper bounds

In this section we show a selection of problems to admit polynomial boundaried kernelization. Our algorithms will be described closely to the (regular) polynomial kernels for the respective problems, which meets the expectation for many reduction rules to be applicable and helpful for boundaried kernelization due to their rather local nature. This will be formally done by showing *gluing safeness* of such rules, i.e., preserving gluing-equivalence. As given by Lemma 3 we assume to be given a boundaried graph G_B such that $G - B$ is either an independent set or a forest, depending on the parameterization. We denote the vertex set of $G - B$ by R.

4.1 Vertex Cover[vc]

Our result for VERTEX COVER[vc] is based on the crown reduction rule, which was first introduced by Chen et al. [14]. A *crown* in a graph G is a pair of vertex sets $I, H \subseteq G$, such that: (i) I is a non-empty independent set of G; (ii) H is the neighborhood of I; and (iii) there exists an H-saturating matching between I and H. Abu-Khzam et al. [1] gave on overview of the technique.

Lemma 6 ([1], Theorem 3)**.** *If G is a graph with crown (I, H), then there is a minimum vertex cover of G, containing all H-vertices and none from I.*

Observe that crowns in a boundaried graph G_B might become invalid after gluing to some boundaried graph F_B, e.g., I-vertices contained in B might become adjacent. For this reason we need to make sure that the intersection of I and B is empty. Furthermore we cannot simply remove vertices of B, as they will be re-introduced by gluing. Instead, we fix an H-saturating matching between I and H, and remove all other edges incident with H. As (I, H) remains a crown after that operation, we mark vertices in I and H to prevent multiple handling, therefore initialize the set of marked vertices $\tilde{I}, \tilde{H} = \emptyset$. Additionally, we remove any isolated vertices in R, as even after gluing, they will not be incident with any edge.

Reduction Rule 1. *If $v \in R$ is an isolated vertex, remove it.*

Proof of gluing safeness. Assume the condition for the rule holds, i.e., $v \in R$ is an isolated vertex. Let $G'_B = G_B - v$ be the resulting graph. For any boundaried graph F_B we know that v is an isolated vertex in $G_B \oplus F_B$, as it is not part of the boundary and thus does not get any new neighbors after gluing. It holds that $\mathrm{OPT}_{\mathrm{VC}}(G_B \oplus F_B) = \mathrm{OPT}_{\mathrm{VC}}(G'_B \oplus F_B)$. □

Reduction Rule 2. *If $G - (\tilde{I} \cup \tilde{H})$ contains a crown (I, H) with $I \subseteq R$ and an H-saturating matching M between H and I, then remove all edges incident with H but not contained in M, and add all vertices in I, H to $\tilde{I}, \tilde{H}$, respectively.*

Proof of gluing safeness. Let (I, H) be a crown that satisfies the above conditions, let G'_B be the resulting graph after removing all edges incident with H, but not contained in M. For any B-boundaried graph F_B the pair (I, H) remains a crown in $G_B \oplus F_B$ since no vertex in I gets new neighbors after gluing, and consequently I remains an independent set, H its neighborhood, and M an H-saturating matching between H and I. Clearly, removing all edges incident with H besides those in M does not change any of those points, and (I, H) is also a crown in $G'_B \oplus F_B$. Also note that, since the only difference between G and G' is about the edges of H, we have $(G_B \oplus F_B) - (I \cup H) = (G'_B \oplus F_B) - (I \cup H)$. Due to Lemma 6, we then get

$$\begin{aligned}\mathrm{OPT}_{\mathrm{VC}}(G_B \oplus F_B) &= \mathrm{OPT}_{\mathrm{VC}}((G_B \oplus F_B) - (I \cup H)) + |H| \\ &= \mathrm{OPT}_{\mathrm{VC}}((G'_B \oplus F_B) - (I \cup H)) + |H| \\ &= \mathrm{OPT}_{\mathrm{VC}}(G'_B \oplus F_B)\end{aligned}$$

□

Such restrictive handling of crowns still suffices in order to bound the independent set number of $G[R]$, as will be given by Lemma 13. Next we show how to apply Reduction Rules 1 and 2 exhaustively in polynomial time by modification of the algorithm by Iwata et al. [33] for removing the crowns in a graph.

First, let $\overline{G}$ be a bipartite graph, which has as vertex set the disjoint union of $\overline{X} \coloneqq \{x_v \mid v \in R \setminus (\tilde{I} \cup \tilde{H})\}$ and $\overline{Y} \coloneqq \{y_v \mid v \in V(G) \setminus (\tilde{I} \cup \tilde{H})\}$, and the edge set is $\{\{x_u, y_v\} \mid \{u, v\} \in E(G - (\tilde{I} \cup \tilde{H})\}$. We refer to the total vertex set of $\overline{G}$ by $\overline{V}$. For vertex set $\overline{S} \subseteq \overline{V}$ we write S_X for $\{v \in R \mid x_v \in \overline{S} \cap \overline{X}\}$ and S_Y for $\{v \in V \mid y_v \in \overline{S} \cap \overline{Y}\}$. The other way around, for vertex set $T \subseteq V$, we define $\overline{X}_T \coloneqq \{x_v \in \overline{X} \mid v \in T\}$ and $\overline{Y}_T \coloneqq \{y_v \in \overline{Y} \mid v \in T\}$. Using $\overline{G}$, we build our actual auxiliary graph $\overline{G}_M$ using a maximum matching M in $\overline{G}$. The vertex set is also $\overline{V}$, but the edges are directed copies of those in $\overline{G}$, directed from $\overline{X}$ to $\overline{Y}$. Only edges belonging to M are additionally directed the other way around, i.e., from $\overline{Y}$ to $\overline{X}$.

At this point, we will also be working with directed graphs, so let us denote, with F being a directed graph and $A \subseteq V(F)$, by $N_F^{\mathtt{out}}(A)$ the set of vertices in $V(F) \setminus A$, to which there is an incoming edge going from some vertex in A.

We show that finding a crown (I, H) in $G - (\tilde{I} \cup \tilde{H})$ with $I \subseteq R$ is equivalent to finding a tail strongly connected component in $\overline{G}_M$, i.e., a set $\overline{S}$ of vertices such that $N^{\mathtt{out}}_{\overline{G}_M}(\overline{S}) = \emptyset$ and $\overline{G}_M[\overline{S}]$ contains directed paths between any two vertices in any direction.

Lemma 7 (Analogue of [33, Lemma 4.5])**.** *If $\overline{S}$ is a tail strongly connected component of $\overline{G}_M$ with $S_X \neq \emptyset$ and $S_X \cap S_Y = \emptyset$, then (S_X, S_Y) is a crown in $G - (\tilde{I} \cup \tilde{H})$ with $S_X \subseteq R$.*

Proof. Since $S_X \subseteq R$ and $(S_X \cup S_Y) \cap (\tilde{I} \cup \tilde{H}) = \emptyset$ follows by definition, we only need to show that (S_X, S_Y) is a crown. Therefore, we simply go through the three conditions of the crown definition.

1. S_X is a non-empty independent set.
   S_X is non-empty by condition of the lemma, and there might be no $u, v \in S_X$ with $\{u, v\} \in E(G)$, as otherwise it would hold that $\{x_u, y_v\} \in \overline{E}$ and thus either $S_X \cap S_Y \neq \emptyset$ or $\overline{S}$ has outgoing edges and is thus not a tail strongly connected component.

2. $S_Y = N_G(S_X)$.
   As $\overline{S}$ is strongly connected, it must hold in particular that $\overline{S} \cap \overline{Y} \subseteq N^{\mathtt{out}}_{\overline{G}_M}(\overline{S} \cap \overline{X})$, and as $\overline{S}$ has no outgoing edges, $N^{\mathtt{out}}_{\overline{G}_M}(\overline{S} \cap \overline{X}) \subseteq \overline{S} \cap \overline{Y}$. Together we get $N^{\mathtt{out}}_{\overline{G}_M}(\overline{S} \cap \overline{X}) = \overline{S} \cap \overline{Y}$ and, since no vertex outside of $\tilde{I} \cup \tilde{H}$ is adjacent to $\tilde{I} \cup \tilde{H}$, also $N(S_X) = S_Y$.

3. There is an S_Y-saturating matching into S_X.
   Since $\overline{S}$ is strongly connected, there is an edge from each $y_v \in \overline{S} \cap \overline{Y}$ to some $x_u \in \overline{S} \cap \overline{X}$ in $\overline{G}_M$, which means $\{y_v, x_u\}$ is contained in the matching M of $\overline{G}$ which was used to generate $\overline{G}_M$. Furthermore, by construction it holds that $\overline{G}[\overline{S}]$ is a subgraph of $G[S_X \cup S_Y]$ and thus the edge set $\{\{u, v\} \mid x_u, y_v \in \overline{S}$ and $\{x_u, y_v\} \in M\}$ is an S_Y-saturating matching into S_X. □

Lemma 8. *Let (I, H) be a crown in $G - (\tilde{I} \cup \tilde{H})$ with $I \subseteq R$, such that $G[I \cup H]$ is connected. If $\overline{Y}_H$ is matched by M into $\overline{X}_I$, then there is a tail strongly connected component $\overline{S} \subseteq \overline{X}_I \cup \overline{Y}_H =: \overline{Q}$ in $\overline{G}_M$, such that $S_X \neq \emptyset$ and $S_X \cap S_Y = \emptyset$.*

Proof. It is well known, that every finite directed graph has at least one tail strongly connected component. Let $\overline{S}$ be any tail strongly connected component of $\overline{G}_M[\overline{Q}]$. Note that $\overline{S}$ remains a tail strongly connected component of $\overline{G}_M$, as $\overline{Q}$, and thus in particular also $\overline{S}$, has no outgoing edges in $\overline{G}_M$: Every vertex in $\overline{Y}_H$ is matched to some vertex in $\overline{X}_I$, which implies $N^{\mathtt{out}}_{\overline{G}_M}(\overline{Y}_H) \subseteq \overline{X}_I$, and since (I, H) is a crown, it holds that $N(I) = H$, which implies $N^{\mathtt{out}}_{\overline{G}_M}(\overline{X}_I) = \overline{Y}_H$.

$S_X \cap S_Y = \emptyset$ is easy to see, as it holds by construction, that $S_X \subseteq I$ and $S_Y \subseteq H$, while $I \cap H = \emptyset$, since I is an independent set and $H = N(I)$. Let us show that $S_X \neq \emptyset$. Assume for contradiction that $\overline{S} \subseteq \overline{Y}$ and thus $S_X = \emptyset$. As seen above, every vertex $y_v \in \overline{S} \cap \overline{Y} \subseteq \overline{Y}_H$ is matched by M to some $x_u \in \overline{X}_I$, and by construction of $\overline{G}_M$ exists an edge from y_v to x_u. As $\overline{S}$ has no outgoing edges, x_u needs to be contained in $\overline{S}$, and we get $S_X \neq \emptyset$. □

Lemma 9. *Let (I, H) be a crown in $G - (\tilde{I} \cup \tilde{H})$ with $I \subseteq R$, such that $G[I \cup H]$ is connected. If $\overline{Y}_H$ is not matched by M into $\overline{X}_I$, then each vertex $y_v \in \overline{Y}_H$ that is not matched by M to any $\overline{X}_I$-vertex, is instead matched by M to some vertex in $\overline{X} \setminus \overline{X}_I$. Furthermore, for each such $y_v \in \overline{Y}_H$ exists some $x_u \in \overline{X}_I$ that is not matched by M, and such that there is an M-alternating path from y_v to x_u, whose first and last edges are not contained in M.*

Proof. Fix any $y_v \in \overline{Y}_H$ which is not matched by M to any $\overline{X}_I$-vertex. Since $H = N(I)$ has a saturating matching into I, and hence $\overline{Y}_H = N_{\overline{G}}(\overline{X}_I)$ has one into $\overline{X}_I$, but M does not match y_v to any vertex in $\overline{X}_I$, it must hold, as M is maximum, that y_v is matched by M to some vertex outside of $\overline{X}_I$.

Construct vertex sets $\overline{Y}_n \subseteq \overline{Y}_H$ and $\overline{X}_n = N_{\overline{G}}(\overline{Y}_n) \cap \overline{X}_I$ as follows. With $\overline{Y}_1 = \{y_v\}$, repeat for $i = 1$ to n such that $\overline{Y}_{n+1} = \overline{Y}_n$: let $\overline{X}_i = N_{\overline{G}}(\overline{Y}_i) \cap \overline{X}_I$ and $\overline{Y}_{i+1} = \overline{Y}_i \cup N_M(\overline{X}_i)$. By Hall's theorem, and since $\overline{Y}_H$ has a saturating matching into $\overline{X}_I$, it holds that $|\overline{Y}_n| \leq |\overline{X}_n|$. At the same time, y_v is not matched to any vertex in $\overline{X}_n \subseteq \overline{X}_I$, so there is at least one vertex $x_u \in \overline{X}_n$ which is not matched by M. By construction of $\overline{X}_n$, there is an M-alternating path from y_v to x_u, with the edges in both ends not contained in M. □

Lemma 10 (Analogue of [33, Lemma 4.6])**.** *If one cannot apply Reduction Rule 1 and there is a crown in $G - (\tilde{I} \cup \tilde{H})$ with $I \subseteq R$, then there exists a tail strongly connected component $\overline{S}$ of $\overline{G}_M$ with $S_X, S_Y \neq \emptyset$ and $S_X \cap S_Y = \emptyset$.*

Proof. We can assume that $G[I \cup H]$ is connected. Otherwise, choose any connected component T of $G[I \cup H]$, and let $I' = I \cap T$ and $H' = H \cap T$. It clearly holds that $H' = N(I')$ and that there exists an H'-saturating matching into I'. Since we cannot apply Reduction Rule 1, it also holds that $H' \neq \emptyset$. Thus, (I', H') is a crown in $G - (\tilde{I} \cup \tilde{H})$ with $I' \subseteq R$, such that $G[I' \cup H']$ is connected.

If $\overline{Y}_H$ is matched by M into $\overline{X}_I$, then we can apply Lemma 8 and are thus done. Hence, assume that $\overline{Y}_H$ is not matched by M into $\overline{X}_I$. Construct as follows a vertex set $\overline{Q} = \overline{X}_n \cup \overline{Y}_n$ with $n \in \mathbb{N}$, such that $\overline{Q}$ induces a (weakly) connected graph in $\overline{G}_M$ and has no outgoing edges outside of itself. Set $\overline{X}_1 = \overline{X}_I$ and repeat for $i = 1$ to n, such that $\overline{X}_{n+1} = \overline{X}_n$: let $\overline{Y}_i = N_{\overline{G}}(\overline{X}_i)$ and $\overline{X}_{i+1} = \overline{X}_i \cup N_M(\overline{Y}_i)$. Observe, that every $y_v \in \overline{Y}_n$ is matched by M to some vertex in $\overline{X}_n$: Assume for contradiction that some $y_v \in \overline{Y}_n$ is not matched by M to some vertex in $\overline{X}_n$. Then by construction of $\overline{X}_n$, it must hold, that y_v is not matched at all by M. By Lemma 9 and maximality of M it cannot hold that $y_v \in \overline{Y}_H$, so we can assume that $y_v \in \overline{Y}_n \setminus \overline{Y}_H$. By construction of $\overline{Y}_n$ it holds that y_v has an M-alternating path to some $y_w \in \overline{Y}_1 = \overline{Y}_H$ which is matched outside of $\overline{X}_I$. Again by using Lemma 9, it then holds that y_v has an M-alternating path, through y_w, to some $x_u \in \overline{X}_I$ with both y_v and x_u not being matched by M, which contradicts maximality of M. Thus, it holds that $N^{\mathtt{out}}_{\overline{G}_M}(\overline{Y}_n) \subseteq \overline{X}_n$. Since it also holds that $N^{\mathtt{out}}_{\overline{G}_M}(\overline{X}_n) = N_{\overline{G}}(\overline{X}_n) = \overline{Y}_n$, we know that $\overline{Q}$ has no outgoing edges in $\overline{G}_M$. Since $\overline{G}[\overline{X}_I \cup \overline{Y}_H]$ is connected, and thus, by construction, also $\overline{G}[\overline{X}_n \cup \overline{Y}_n]$, it holds that $\overline{G}_M[\overline{Q}]$ is connected.

It is well known that every finite directed graph has a tail strongly connected component. Let $\overline{S}$ be any such component of $\overline{G}_M[\overline{Q}]$. As observed earlier, every $y_v \in \overline{S} \cap \overline{Y}_n$ has an outgoing edge to some vertex in

$\overline{X}_n$. On the other hand, since we cannot apply Reduction Rule 1 and $\overline{Y}_n = N^{\mathtt{out}}_{\overline{G}_M}(\overline{X}_n)$, every $x_u \in \overline{S} \cap \overline{X}_n$ has an outgoing edge to some vertex in $\overline{Y}_n$. This way, $\overline{S}$ contains at least one $\overline{X}$ and at least one $\overline{Y}$ vertex, i.e., it holds that $S_X, S_Y \neq \emptyset$.

It remains to show that $S_X \cap S_Y = \emptyset$. Assume for contradiction that there is some $v \in S_X \cap S_Y$, i.e., it holds that $x_v \in \overline{S} \cap \overline{X}$ and $y_v \in \overline{S} \cap \overline{Y}$. Let $u_1 = v$ and repeat from $i = 1$ to k, such that $u_{k+1} \in \{u_1, \ldots, u_k\}$: choose u_{i+1} such that $x_{u_{i+1}}$ is the M-neighbor of y_{u_i}. Thus, u_{i+1} is a neighbor of u_i in G, and as a result, $y_{u_{i+1}}$ is an outgoing neighbor of x_{u_i}. As $\overline{S}$ is strongly connected, we have visited all its vertices until $i = k$, so it holds that $\overline{S} = \{x_{u_1}, y_{u_1}, \ldots, x_{u_k}, y_{u_k}\}$. Furthermore, it holds that $\overline{S} = \overline{Q}$, as otherwise by connectedness of $\overline{G}_M[\overline{Q}]$ there needs to be some edge e between $\overline{Q} \setminus \overline{S}$ and $\overline{S}$. If e is outgoing from $\overline{S}$, we contradict the fact that $\overline{S}$ is a tail strongly connected component. Else e is an ingoing edge for $\overline{S}$, and since for every vertex in $\overline{S} \cap \overline{X}$ its only ingoing neighbor is also contained in $\overline{S}$, this edge goes from some $x_p \in \overline{Q} \setminus \overline{S}$ to some $y_q \in \overline{S}$. But then there also exists the edge from $x_q \in \overline{S}$ to $y_p \in \overline{Q} \setminus \overline{S}$, which again contradicts the fact that $\overline{S}$ has no outgoing edges. From $\overline{S} = \overline{Q}$ it follows that $\overline{X}_I \subseteq \overline{S}$, which together with $\overline{S}$ being strongly connected implies that all $\overline{X}_I$-vertices are matched by M. This leads to a contradiction: By earlier assumption, M does not match $\overline{Y}_H$ into $\overline{X}_I$; and at the same time, as (I, H) is a crown, it holds that $|\overline{X}_I| \geq |\overline{Y}_H|$ and that $\overline{X}_I$ has no neighbors outside of $\overline{Y}_H$, implying that not all $\overline{X}_I$-vertices are matched by M. □

Lemma 11. *Reduction Rules 1 and 2 can be exhaustively applied in polynomial time.*

Proof. First, remove all isolated vertices contained in R. Then, set $\tilde{I}, \tilde{H} = \emptyset$ and repeatedly apply Reduction Rule 2 by constructing $\overline{G}_M$ and finding a tail strongly connected component $\overline{S}$ in $\overline{G}_M$ with $S_X \cap S_Y = \emptyset$ and $S_X, S_Y \neq \emptyset$, for which, by Lemma 7, it holds that (S_X, S_Y) is a crown in $G - (\tilde{I} \cup \tilde{H})$ with $S_X \subseteq R$. If $E(S_Y, R)$ is more than a matching between S_Y and S_X, delete the superfluous edges. After each such step, we set $\tilde{I} = \tilde{I} \cup S_X$, $\tilde{H} = \tilde{H} \cup S_Y$ and remove any new isolated vertices in R. Furthermore, the repeated computation of $\overline{G}_M$ can be avoided by simply removing $\overline{S}$ from $\overline{G}_M$, since S_X and S_Y are now part of $\tilde{I}$ and $\tilde{H}$. By Lemma 10, if we cannot apply Lemma 7, then there are no crowns left in $G - (\tilde{I} \cup \tilde{H})$ with $I \subseteq R$. □

Lemma 12. *If there is a crown (I, H) in (G, B) with $I \subseteq R$ and $|I| > |H|$, then we can apply one among Reduction Rules 1 and 2.*

Proof. From the condition that (I, H) is a crown it follows that $H = N(I)$ and there exists an H-saturating matching M between H and I. If $E(H, I) \setminus M \neq \emptyset$, we apply Reduction Rule 2. Otherwise, we get $|I| > |H| = |M| = |E(H, I)|$ and thus at least one vertex in $I \subseteq R$ is isolated, so we apply Reduction Rule 1. □

Now that we can remove all isolated vertices and crowns (I, H) with $I \subseteq R$ and $|I| > |H|$ in polynomial time, we state that the resulting graphs do not have large independent sets. Since for VERTEX COVER[vc] all of R is an independent set, it thus cannot be larger than B, which gives us the boundaried kernel. Note that neither the reduction rules, nor the polynomial-time application of them, nor the following lemma expect $G[R]$ to be an independent set, making this lemma also useful for other structural parameterization.

Lemma 13. *If $\alpha(G[R]) > \frac{1}{2}|V(G)|$, then we can apply one among Reduction Rules 1 and 2.*

Proof. Let Y be a maximum independent set of $G[R]$, i.e., such that $|Y| = \alpha(G[R])$. Let X be the neighborhood of Y, which is clearly a subset of $V(G) \setminus Y$. Note that since $|Y| = \alpha(G[R]) > \frac{1}{2}|V(G)|$, it thus holds that $|X| \leq |V(G)| - |Y| < \frac{1}{2}|V(G)| < |Y|$.

If there exists an X-saturating matching into Y, then by Lemma 12 we can apply one of the reduction rules. Else we choose some maximal $\hat{X} \subseteq X$ such that $|\hat{X}| > |\hat{Y}|$ with $\hat{Y} = N(\hat{X}) \cap Y$, which exists by Hall's Theorem. Let $X^* = X \setminus \hat{X}$ and $Y^* = Y \setminus \hat{Y}$. Note that even though X^* might have neighbors in $\hat{Y} = Y \setminus Y^*$, it holds that $N(Y^*) \subseteq X^*$, and by maximality of $\hat{X}$, also $N(Y^*) = X^*$. Further, as $Y^* \neq \emptyset$, if (Y^*, X^*) is not a crown, it can only be due to absence of an X^*-saturating matching into Y^*. Then, again by Hall's Theorem, there exists some $\bar{X} \subseteq X^*$ with $|\bar{X}| > |N(\bar{X}) \cap Y^*|$, which contradicts maximality of $\hat{X}$. As a result, (Y^*, X^*) must be a crown with $|Y^*| = |Y| - |\hat{Y}| > |Y| - |\hat{X}| > |X| - |\hat{X}| = |X^*|$. Again, by Lemma 12 we can apply one of Reduction Rules 1 and 2. □

Theorem 4. *The parameterized problem* VERTEX COVER[vc] *admits a polynomial boundaried kernelization with at most* $2(|B| + k)$ *vertices.*

Proof. By Lemma 3 assume that we are given boundaried graph G_B with $G - B$ an independent set. Using Lemma 11 apply Reduction Rules 1 and 2 exhaustively in polynomial time, resulting in some boundaried graph G'_B. Observe that G' is a subgraph of G, implying that $R' := V(G') \setminus B$ is an independent set, leading to $|R'| = \alpha(G'[R']) \leq \frac{1}{2}|V(G')|$ by Lemma 13 and thus $|V(G')| = |B| + |R'| \leq 2|B|$. As a result, we can output G'_B with $G' - B$ an independent set and $\Delta = 0$. □

4.2 Vertex Cover[fvs]

Using the polynomial kernel for VERTEX COVER[fvs] by Bodlaender and Jansen [34] as a base, we describe our polynomial boundaried kernel for this problem. Main steps of the kernel by Bodlaender and Jansen are to make sure that $G - X$ (with X a forest-modulator) has a perfect matching, to reduce the number of so called conflict structures in $G - X$, and to use that in order to bound the total size of R. We use Lemma 13 for the first step, as it allows us to move vertices that are problematic for a perfect matching in $G[R]$ to B, with only a linear blow-up of the latter. Regarding the last two steps, we can mainly apply the same reduction rules as Bodlaender and Jansen. We only need to be careful to not delete boundary-vertices. Instead, we mark them by giving personal leaves and thus forcing them into a solution.

Lemma 14 (Analogue of [34, Lemma 1])**.** *Let* $G_B \in \mathcal{C}_B^{forest}$ *such that Reduction Rules 1 and 2 cannot be applied. In polynomial time one can compute a set* $B' \supseteq B$ *of size at most* $2|B|$*, such that* $G - B'$ *has a perfect matching.*

Proof. With the Hopcroft-Karp algorithm we find a maximum matching M of the forest $G[R]$, and let I be the (independent) set of vertices not covered by M. Note that this way it holds that $|R| = 2|M| + |I|$. Since $G[R]$ is a forest and thus bipartite, by König's Theorem it has a minimum vertex cover of size $|M|$, and thus a maximum independent set of size $|R| - |M| = |M| + |I|$. From Lemma 13 it follows that $|M| + |I| = \alpha(G[R]) \leq 1/2|V(G)| = 1/2(|B| + 2|M| + |I|)$, and hence $|I| \leq |B|$. For B' we can thus choose the set $B \cup I$. □

Next, let us use additional definitions, which were first introduced by Bodlaender and Jansen [34] and generalized by Hols et al. [32].

Definition 2 (Blocking sets)**.** *For a graph* G*, a vertex set* $Y \subseteq V(G)$ *is called a blocking set in* G*, if no minimum vertex cover of* G *contains* Y*, i.e., for all solutions* S *with* $Y \subseteq S$ *it holds that* $|S| > \mathrm{OPT}_{VC}(G)$*. A blocking set* Y *is called a minimal blocking set, if no strict subset of* Y *is also a blocking set.*

We denote by $\beta(G)$ *the size of the largest minimal blocking set in* G*. Further, for graph class* $\mathcal{C}$ *let* $b_{\mathcal{C}} := \max_{G \in \mathcal{C}} \beta(G)$ *with* $b_{\mathcal{C}} = \infty$ *if the minimal blocking set size of graphs in* $\mathcal{C}$ *can be arbitrarily large.*

Definition 3 (Chunks and conflicts)**.** *Let* G *be a graph,* $\mathcal{C}$ *a graph class and* X *a* $\mathcal{C}$*-modulator for* G*. A chunk in* X *is an independent vertex set* $Z \subseteq X$ *of size at most* $b_{\mathcal{C}}$*. If* X *is clear from context, we denote the set of chunks in* X *as* $\mathcal{X}$*.*

With $F \subseteq V(G) \setminus X$*, and* $Z \in \mathcal{X}$*, let* $\mathrm{CONF}_F(Z) := \mathrm{OPT}_{VC}(G[F] - N_F(X)) + |N_F(X)| - \mathrm{OPT}_{VC}(G[F])$ *be the conflict caused by* Z *on* F*.*

Note that Bodlaender and Jansen have shown that 2 is the maximum size of a minimal blocking set in a forest, i.e., it holds that $b_{\mathcal{C}^{\mathtt{forest}}} = 2$ [34, Lemma 3]. Thus, let $\mathcal{X} := \{Z \subseteq B \mid G[Z]$ is an independent set and $|Z| \leq 2\}$ be the chunk-set for G_B. The following reduction rule is easily seen to be gluing safe, as it works with components in $G[R]$ for which we can take the locally optimum solution, without needing to worry about possible conflicts after gluing. This rule can be seen as a special case of Reduction Rule 15.

Reduction Rule 3. *If* $G[R]$ *contains a connected component* T *such that for every chunk* $Z \in \mathcal{X}$ *it holds that* $\mathrm{CONF}_T(Z) = 0$*, then delete* T *and increase* Δ *by* $\mathrm{OPT}(T)$*.*

Next, let us bound the total number of conflicts that chunks in $\mathcal{X}$ can cause on R.

Lemma 15. *If for some chunk $Z \in \mathcal{X}$ it holds that $\text{CONF}_{G[R]}(Z) \geq |B|$, then for every H_B there exists an optimum vertex cover S of $G_B \oplus H_B$ such that $Z \cap S \neq \emptyset$.*

Proof. Let H_B be fixed and let S be a minimum vertex cover of $G_B \oplus H_B$. If it holds that $Z \cap S \neq \emptyset$, then we are done. So assume the opposite, i.e., that $Z \cap S = \emptyset$. For each connected component T of $G[R]$, fix some minimum vertex cover S_T of T. Let $\hat{S}$ be the union of all such S_T. Construct the set $S' \coloneqq (S \setminus R) \cup B \cup \hat{S}$. It holds that S' is a vertex cover of $G_B \oplus H_B$: (i) all edges in H are covered, since $S' \cap V(H) \supseteq S \cap V(H)$; (ii) all edges in $G[R]$ are covered, since S' contains all vertices in $\hat{S}$; (iii) all edges between R and B are covered, as $B \subseteq S'$. Since we assumed that $Z \cap S = \emptyset$ and it holds that $\text{CONF}_{G[R]}(Z) \geq |B|$, it follows that $|S'| \leq |S| - |B| + |B| = |S|$, and thus S' is also a minimum vertex cover of $G_B \oplus H_B$. □

Essentially, Bodlaender and Jansen used Lemma 15 to remove any chunk of size one with conflict at least $|B|$ in R, and, if the chunk has size two instead, to add an edge between the two vertices of Z. In our case, however, we cannot simply remove vertices in the boundary B, as they would get reintroduced after gluing. Instead, we mark such vertices by giving them special leaves. This is safe due to the well known reduction rule for VERTEX COVER, that neighbors of leaves can be safely taken to the solution.

Let $L = \{l_x \in V(G) \mid x \in B\}$ be the set of special leaves of boundary-vertices. At the beginning we set $L = \emptyset$. From now on, we set $R = V(G) \setminus (B \cup L)$. Note that at any point it will hold that $|L| \leq |B|$.

Reduction Rule 4. *If there is a vertex $x \in B$ without leaf $l_x \in L$, and such that $\text{CONF}_R(x) \geq |B|$, then (i) add l_x to L with the edge $\{x, l_x\}$; and (ii) delete all edges between x and R.*

Proof of gluing safeness. Let G'_B be the resulting graph. Fix any B-boundaried graph H_B. Let S' be a minimum vertex cover for $G'_B \oplus H_B$. Since x is adjacent to a leaf l_x in $G'_B \oplus H_B$, we can assume that x is contained in S' (else $l_x \in S'$ and we can exchange it with x). Thus, all edges which exist in $G_B \oplus H_B$, but not in $G'_B \oplus H_B$ (namely, those between x and R), are all covered by S'.

Now let S be a minimum vertex cover for $G_B \oplus H_B$. By Lemma 15 we can assume that S contains x. Thus, we know that S also covers the only edge in $G'_B \oplus H_B$ that is not contained in $G_B \oplus H_B$ (namely $\{x, l_x\}$), and hence S is also a minimum vertex cover for $G'_B \oplus H_B$. □

Reduction Rule 5. *If there is a chunk $\{x, y\} \in \mathcal{X}$ with $\text{CONF}_R(\{x, y\}) \geq |B|$, then add an edge between x and y, and remove $\{x, y\}$ from the chunk set $\mathcal{X}$.*

Proof of gluing safeness. Let G'_B be the resulting graph. For any H_B it holds that $G_B \oplus H_B$ is a subgraph of $G'_B \oplus H_B$, so $\text{OPT}_{\text{VC}}(G_B \oplus H_B) \leq \text{OPT}_{\text{VC}}(G'_B \oplus H_B)$ is obvious.

Now let S be a minimum vertex cover for $G_B \oplus H_B$. By Lemma 15 we can assume that S contains x or y. Thus, we know that S also covers the only edge in $G'_B \oplus H_B$ that is not contained in $G_B \oplus H_B$ (namely $\{x, y\}$), and hence S is also a minimum vertex cover for $G'_B \oplus H_B$. □

As a last step, we need to bound the number of such local structures, on which the chunks do not cause any conflict, i.e., we know that any minimum vertex cover for the global graph (even after gluing) is also optimal locally. For this, we use the notion of blockability which was also introduced by Bodlaender and Jansen.

Definition 4. *The vertex-pair $u, v \in R$ is called blockable, if there is a chunk $Z \in \mathcal{X}$ such that $u, v \in N(Z)$.*

Vertices u, v being blockable by some chunk Z can be seen in the sense that forbidding Z from a vertex cover solution actually implies that we are forced to take u or v into the solution. It follows from the definition, that if u, v are not blockable, then for any two $x \in N(u) \cap B$ and $y \in N(v) \cap B$ it holds that $x \neq y$ and $\{x, y\} \in E(G)$, i.e., no chunk $Z \in \mathcal{X}$ is adjacent to both x and y.

Reduction Rule 6. *If $u, v \in R$ are not blockable and $\deg_{G[R]}(u), \deg_{G[R]}(v) \leq 2$, then do the following: (i) delete u and v, and increase Δ by 1; (ii) if u has a neighbor $t \neq v$ in R, then make it adjacent to the old B-neighborhood of v; (iii) if v has a neighbor $w \neq u$ in R, then make it adjacent to the old B-neighborhood of u; (iv) if both t and w exist, then add the edge $\{t, w\}$.*

Proof of gluing safeness. Let G' be the modified graph and fix some H_B. We show that $\mathrm{OPT}(G_B \oplus H_B) = \mathrm{OPT}(G'_B \oplus H_B)+1$. Since we work with two graphs at the same time, namely those are $G_B \oplus H_B$ and $G'_B \oplus H_B$, let us fix that $N_B(u)$, resp., $N_B(v)$, is used for $N_{G_B \oplus H_B}(u) \cap B = N_G(u) \cap B$, resp., $N_{G_B \oplus H_B}(v) \cap B = N_G(v) \cap B$.

($\mathrm{OPT}(G_B \oplus H_B) \geq \mathrm{OPT}(G'_B \oplus H_B) + 1$) Let us first note the differences in conditions that a vertex cover S' for $G'_B \oplus H_B$ needs to fulfill, compared to what conditions a minimum vertex cover S for $G_B \oplus H_B$ fulfills. That is, S' does not need to cover any edges incident with u or v, but it needs to cover: (i) the edges between t and $N_B(v)$, if t exists; (ii) the edges between w and $N_B(u)$, if w exists; and (iii) the edge $\{t, w\}$, if they both exist.

Next, let us show that we can assume without loss of generality that $N_B(u) \cup \{v\} \subseteq S$. There is an edge between u and v, so $S \cap \{u, v\} \neq \emptyset$. As u, v are not blockable, there are no common vertices in $N_B(u)$ and $N_B(v)$, and every vertex in $N_B(u)$ is adjacent to every vertex in $N_B(v)$, so either one needs to be fully contained in S. The case $N_B(v) \cup \{u\} \subseteq S$ is symmetrical to $N_B(u) \cup \{v\} \subseteq S$. The cases where neither $N_B(u) \subseteq S$ nor $u \in S$, resp., neither $N_B(v) \subseteq S$ nor $v \in S$, trivially lead to uncovered edges.

Now, if t exists, we can assume w.l.o.g. that $t \in S$. Otherwise it holds that $u \in S$ and thus $(S \setminus \{u\}) \cup \{t\}$ is also a minimum vertex cover of $G_B \oplus H_B$. Hence, all new edges incident with t are automatically covered by $S' \coloneqq S \setminus \{v\}$. If w exists, then either $w \in S$ or $N_{G_B \oplus H_B}(w) \subseteq S$. In the first case, all new edges incident with w are automatically covered by S'. And the same also holds in the second case, since we assumed w.l.o.g. that $N_B(u) \subseteq S$ and $t \in S$.

($\mathrm{OPT}(G_B \oplus H_B) \leq \mathrm{OPT}(G'_B \oplus H_B) + 1$) Again, let us first observe the new constraints on a vertex cover S for $G_B \oplus H_B$, as compared to the constraints on a minimum vertex cover S' for $G'_B \oplus H_B$. Those are that S additionally needs to cover the edge $\{u, v\}$, the edges between u and $N_B(u)$ and between v and $N_B(v)$, and the edges $\{t, u\}, \{v, w\}$ if t, resp., w, exist.

Since $N_B(u) \cup N_B(v)$ induces a complete bipartite graph on $G'_B \oplus H_B$, we can assume w.l.o.g. that $N_B(u) \subseteq S'$. Let $S = S' \cup \{v\}$ and note that S covers $\{u, v\}$ and the edges between $N_B(u)$ and u, and between $N_B(v)$ and v. If neither t nor w exist, we are done by choosing S. If both t and w exist, we can assume w.l.o.g. that $t \in S'$ (note that else it holds that $w \in S'$ and $N_B(v) \subseteq S'$, which is symmetrical to $t \in S'$ and $N_B(u) \subseteq S'$, so we choose $S' \cup \{u\}$), and thus S covers also the new edges $\{t, u\}$ and $\{v, w\}$. If w exists, but t does not, then again S covers the new edge $\{v, w\}$. Else it holds that t exists and w does not. If $t \in S'$, we are done by choosing S. Else it holds that $N_{G'_B \oplus H_B}(t) \subseteq S'$. Hence, in particular, it holds that $N_B(v) \subseteq S'$. In that case we choose $S' \cup \{u\}$ as the vertex cover for $G_B \oplus H_B$, and note that it indeed covers all new edges of t, u and v. □

Reduction Rule 7. *If there are distinct vertices t, u, v, w in R which satisfy $\deg_{G[R]}(u) = \deg_{G[R]}(v) = 3$, $N_R(t) = \{u\}$, $N_R(w) = \{v\}$ and $\{u, v\} \in E(G)$ such that the pairs $\{u, t\}, \{v, w\}$ and $\{t, w\}$ are not blockable, then do the following: (i) delete t, u, v and w, and increase Δ by 2; (ii) with p being the third neighbor of u in R, make p adjacent to the old B-neighborhood of t; (iii) with q being the third neighbor of v in R, make q adjacent to the old B-neighborhood of w.*

Proof of gluing safeness. Let G' be the modified graph and fix some H_B. We show that $\mathrm{OPT}(G_B \oplus H_B) = \mathrm{OPT}(G'_B \oplus H_B) + 2$.

($\mathrm{OPT}(G_B \oplus H_B) \geq \mathrm{OPT}(G'_B \oplus H_B) + 2$) Let S be a minimum vertex cover of $G_B \oplus H_B$. We do a case distinction. Let the first case be that both $u, v \in S$. If it also holds that $N_B(t), N_B(w) \subseteq S$, then $S \setminus \{u, v\}$ automatically covers all new edges in $G'_B \oplus H_B$. Otherwise assume w.l.o.g. that $N_B(w) \not\subseteq S$. Then it must hold that $w \in S$ and $N_B(v) \subseteq S$, as $\{v, w\}$ is not blockable. We then observe that $(S \setminus \{v\}) \cup \{q\}$ is also a vertex cover of same size as S for $G_B \oplus H_B$, and this corresponds to the next case.

Our second case is, w.l.o.g., that $u \in S$ and $v \notin S$. Since $v \notin S$, it follows that $w, q \in S$ and $N_B(v) \subseteq S$. If additionally it holds that $N_B(t) \subseteq S$, then $S \setminus \{u, w\}$ contains q and covers the new edges of p. Otherwise it needs to hold that $t \in S$ and $N_B(u), N_B(w) \subseteq S$, as $\{u, t\}$ and $\{t, w\}$ are not blockable. Then we also know that $(S \setminus \{u, w\}) \cup \{p, v\}$ is a vertex cover of the same size as S for $G_B \oplus H_B$, and this corresponds to the case that $v \in S$, $u \notin S$ and $N_B(w) \subseteq S$, which is symmetrical to $u \in S, v \notin S$ and $N_B(t) \subseteq S$. Thus we are done.

($\mathrm{OPT}(G_B \oplus H_B) \leq \mathrm{OPT}(G'_B \oplus H_B) + 2$) Let S' be a minimum vertex cover of $G'_B \oplus H_B$. Since $\{t, w\}$ are not blockable in G, it holds that $N_B(t) \cup N_B(w)$ induce a complete bipartite graph in G, thus also in G' and $G'_B \oplus H_B$. This way, we determined that it must hold that $N_B(t) \subseteq S'$ or that $N_B(w) \subseteq S'$. W.l.o.g. assume the former. If the latter also holds, then $S' \cup \{u, v\}$ covers all new edges in $G_B \oplus H_B$ which are not contained in $G'_B \oplus H_B$. Namely, those are $\{p, u\}, \{t, u\}, \{u, v\}, \{v, w\}, \{v, q\}$, the edges between u and $N_B(u)$, between v and $N_B(v)$, between t and $N_B(t)$, and between w and $N_B(w)$.

Otherwise it holds that $N_B(w) \not\subseteq S'$, and thus, since q is adjacent to $N_B(w)$ in G', and since w and v are not blockable in G, it must be the case that $q \in S'$ and that $N_B(v) \subseteq S'$. In such case it suffices to choose $S' \cup \{u, w\}$. □

Note that the number of conflicts that a chunk $Z \in \mathcal{X}$ induces on a component in $G[R]$, as well as what structures are allowed to remain in $G[R]$, is exactly the same as in the kernel of Bodlaender and Jansen, since Reduction Rules 4 and 5 still bound the sum of possible conflicts induced by $\mathcal{X}$, and Reduction Rules 3, 6 and 7 are exactly the same (neglecting that we increase the solution size offshift Δ instead of decreasing the sought solution size k).

Definition 5. *Define the number of active conflicts induced on the forest F by the chunks $\mathcal{X}$ as* $\textsc{active}_F(\mathcal{X}) \coloneqq \Sigma_{Z \in \mathcal{X}} \textsc{conf}_F(Z)$.

Lemma 16 ([34], Observation 4)**.** *Let G_B be a boundaried graph such that $G - B$ is a forest. Let L be a set of degree-1 vertices of $G - B$ that are adjacent to B. Assume that $G - (B \cup L)$ has a perfect matching, and that one cannot apply Reduction Rules 3 to 7. Let $R = V(G) \setminus \{B \cup L\}$. Then it holds that* $\textsc{active}_R(\mathcal{X}) \leq |B|^2 + \binom{|B|}{2}|B|$.

Definition 6 (Conflict structures)**.** *Let F be a forest with a perfect matching M.*
A conflict structure of type A in F is a pair of distinct vertices $\{v_1, v_2\}$ such that $\{v_1, v_2\} \in M$ and $\deg_F(v_1), \deg_F(v_2) \leq 2$.
A conflict structure of type B in F is a path on four vertices (v_1, v_2, v_3, v_4) such that v_1 and v_4 are leaves of F, and $\deg_F(v_2) = \deg_F(v_3) = 3$.

Lemma 17 ([34], Lemma 7)**.** *Let G_B be a boundaried graph such that $G - B$ is a forest. Let L be a set of degree-1 vertices of $G - B$ that are adjacent to B. Assume that $G - (B \cup L)$ has a perfect matching, and that one cannot apply Reduction Rules 3 to 7. Let $R = V(G) \setminus \{B \cup L\}$. Let $\mathcal{S}$ be a set of vertex-disjoint conflict structures in R. Then* $\textsc{active}_R(\mathcal{X}) \geq |\mathcal{S}|$.

Lemma 18 ([34], Theorem 1)**.** *Let T be a tree with a perfect matching. There is a set $\mathcal{S}$ of mutually vertex-disjoint conflict structures in T with $|\mathcal{S}| \geq |V(T)|/14$.*

Lemma 19 ([34], Lemma 9)**.** *We can exhaustively apply Reduction Rules 3 to 7 on G_B in polynomial time.*

Lemma 20. *Let G_B be a boundaried graph and $L \subseteq V(G) \setminus B$ a set of degree-1 vertices that are adjacent to B, such that for $R \coloneqq V(G) \setminus (B \cup L)$ it holds that $G[R]$ has a perfect matching. If one cannot apply any of Reduction Rules 3 to 7, then it holds that $|R| \leq 14(|B|^2 + |B|^3)$.*

Proof. Let $\mathcal{S}$ be the set of conflict structures in $G[R]$ which is obtained by repeated use of Lemma 18 on each component of $G[R]$. It holds that $|\mathcal{S}| \geq |R|/14$. By Lemmas 16 and 17 we get $|B'|^2 + \binom{|B|}{2}|B| \geq \textsc{active}_R(\mathcal{X}) \geq |R|/14$, i.e., it holds that $|R| \leq 14(|B|^2 + |B|^3)$. □

Theorem 5. *The parameterized problem* VERTEX COVER[fvs] *admits a polynomial boundaried kernelization with at most $\mathcal{O}((|B| + k)^3)$ vertices.*

Proof. Let boundaried graph G_B and feedback vertex set X of G be given. Using Reduction Rules 1 and 2 and Lemma 14, compute a set $B' \supset B \cup X$ of size at most $2(|B| + k)$, such that $G - B'$ has a perfect matching. Note that $G - B'$ also remains a forest. Using Lemma 19, exhaustively apply Reduction Rules 3 to 7 on $G_{B'}$ in polynomial time. Let $G'_{B'}$ be the resulting graph, L the set of vertices introduced by the

repeated use of Reduction Rule 4, Δ the resulting offset and $R = V(G') \setminus (B' \cup L)$. By Lemma 20 it holds that $|R| \leq 14(|B'|^2 + |B'|^3)$. By gluing safeness of the applied reduction rules, it holds that $G_{B'} \equiv_{\mathrm{VC},B'} G'_{B'}$ and since $B \subseteq B'$, we know by Lemma 1 that for all H_B: $\mathrm{OPT}(G_B \oplus H_B) = \mathrm{OPT}(G'_B \oplus H_B) + \Delta$. As a result, our boundaried kernelization algorithm outputs G'_B with at most $\mathcal{O}((|B|+k)^3)$ vertices, its feedback vertex set B' of size $2(|B|+k)$ and offset Δ. □

4.3 Feedback Vertex Set[fvs]

The state of the art vertex-quadratic kernelization for FEEDBACK VERTEX SET[k] was given by Thomassé [46]. It mainly works through handling vertices of low degrees, as well as certain flowers, i.e., sets of cycles intersecting at exactly one vertex, which are shown to exist at vertices of high degree, and recognizing that this bounds the total number of vertices by a quadratic function on the sought solution size, unless certainly given a NO-instance. We have no sought solution size at hand, but we have a nice structure with $G[R]$ only consisting of trees as components. Using similar reduction rules to those given by Thomassé, we force the low-degree vertices of R (i.e., leaves and non-branching vertices) to be adjacent to B on one hand, and limit the maximum degree of the B-vertices by $\mathcal{O}(|B|)$ on the other hand. Together, this bounds the allowed number of vertices in R by $\mathcal{O}(|B|^2)$.

Throughout this section we allow loops, i.e., an edge from a vertex to itself, and double edges, i.e., that an edge $\{u, v\}$ exists up to twice. If an edge already exists twice and we would try to add a third copy, we instead discard the new copy.

First, let us handle R-vertices of low degree, B-vertices with loops and vertices $x \in B$ that admit an x-flower of order greater than $|B|$, i.e., x is the intersection point of at least $|B|+1$ many otherwise disjoint cycles. The gluing safeness of these rules is easy to see, since R-vertices of degree 0 and 1 are not contained in any cycle, even after gluing; while B-vertices with loops and high-order flowers need to be contained in any solution, hence additional edges at these vertices are of no interest.

Reduction Rule 8. *If exists vertex $v \in R$ with degree 0 or 1, then delete v.*

Reduction Rule 9. *If some vertex $v \in R$ is incident with exactly two edges $\{v, u\}$ and $\{v, w\}$ (possibly with $u = w$), then delete v and add the edge $\{u, w\}$.*

Reduction Rule 10. *If for some vertex $x \in B$ there is a loop at x and x is incident with further edges, then delete all those other edges incident with x.*

Reduction Rule 11. *If for some vertex $x \in B$ there is an x-flower F in G of order $p > |B|$, then add a loop at x and delete all other edges incident with x.*

Next, we work with vertices $x \in B$ that admit a special structure, to which we first want to give a little intuition: Assume that for x there exists a vertex set $X \in V \setminus \{x\}$ and set $\mathcal{C}$ of connected components in $G - (X \cup \{x\})$ not intersecting B, such that there is exactly one edge between x and any component $C \in \mathcal{C}$, and for any subset $Z \subseteq X$ there are at least $2|Z|$ components in $\mathcal{C}$ that are adjacent to Z. The last point implies, by Hall's theorem, that we can assign to each vertex in X its distinct two adjacent components of $\mathcal{C}$. This way, we obtain an x-flower of order $|X|$, in which each cycle goes from x through some component in $\mathcal{C}$ to some vertex in X, and back to x through another component in $\mathcal{C}$, with each cycle visiting its distinct two components. Since each component of $\mathcal{C}$ has no neighbors outside of $X \cup \{x\}$ (even after gluing, as it does not intersect B) and is itself a tree, there exists an optimum solution which contains either x or all of X. We refer to the work of Thomassé [46] for the formal proof of (gluing) safety, since by choice of $V(C) \cap B = \emptyset$ for every $C \in \mathcal{C}$, we make sure that this structure preserves even after gluing.

Reduction Rule 12. *Assume there is a vertex $x \in V$, a vertex set $X \subseteq V \setminus \{x\}$ and a set $\mathcal{C}$ of connected components of $G - (X \cup \{x\})$ (not necessarily all of the connected components) such that: (i) for every $C \in \mathcal{C}$ it holds that $V(C) \cap B = \emptyset$; (ii) there is exactly one edge between x and every $C \in \mathcal{C}$; (iii) every $C \in \mathcal{C}$ induces a tree; (iv) for every subset $Z \subseteq X$, the number of components in $\mathcal{C}$ having some neighbor in Z is at least $2|Z|$. Then add a double edge between x and every vertex in X, and delete the edges between x and the components in $\mathcal{C}$.*

As a last step, we show that as long as some B-vertex has at least $5|B|$ neighbors in R, we can find a reduction rule to apply. Both finding the rule and applying it can be done in polynomial time.

Lemma 21 ([46], Corollary 2.2)**.** *Let G be a graph and x be a vertex of G which is not incident with a loop. The maximum order of an x-flower is equal to the minimum of $|X| + \Sigma_{C\in\mathcal{C}}\lfloor\frac{e(x,C)}{2}\rfloor$, where X is a subset of vertices, $\mathcal{C}$ is the set of components of $G-(X\cup\{x\})$, and $e(x,C)$ is the number of edges between x and vertices in C. Moreover, the set X can be computed in time polynomial in the size of G.*

Lemma 22 ([46], Theorem 2.4)**.** *Let G be a nonempty bipartite graph with bipartition $V = X\dot\cup Y$ with $|Y| \geq 2|X|$ and such that every vertex of Y has at least one neighbor in X. Then there exist nonempty sets $X' \subseteq X$ and $Y' \subseteq Y$ such that $N(Y') = X'$ and such that every $Z \subseteq X'$ has at least $2|Z|$ neighbors in Y'. In addition, X' and Y' can be computed in time polynomial in the size of G.*

Lemma 23. *If for some $x \in B$ it holds that $|N_R(x)| \geq 5|B|$, then one can recognize applicability of one among Reduction Rules 8 to 12 in time polynomial in the size of G.*

Proof. Recognizing Reduction Rules 8 to 10 is simple to do, so assume that none of them can be applied. In particular, there is no loop at x, and thus we make use of Lemma 21 in order to find a set of vertices $X \subseteq V\setminus\{x\}$, such that the maximum order p of an x-flower is equal to $|X| + \Sigma_{C\in\mathcal{C}}\lfloor\frac{e(x,C)}{2}\rfloor$, where $\mathcal{C}$ is the set of components of $G-(X\cup\{x\})$, and $e(x,C)$ is the number of edges between x and C. If $p > |B|$, then apply Reduction Rule 11. Otherwise, let $\mathcal{C}' \subseteq \mathcal{C}$ be the set of components of $\mathcal{C}$ that have more than one edge to x. We denote by e' the total number of edges between x and the components of $\mathcal{C}'$. Note that by choice of X, it holds that $|X| + e'/3$ is at most p. Since we could not apply Reduction Rule 11, there is no x-flower of order $|B|+1$, so we get $3|X| + e' \leq 3p \leq 3|B|$ and thus also $|X| \leq |B|$. Furthermore, there are at most $|B|$-many components in $\mathcal{C}\setminus\mathcal{C}'$ that contain B-vertices.

Together, the number c of components in $\mathcal{C}$ that are disjoint from B and linked to x with exactly one edge is at least as high as the minimum degree $5|B|$ of x, minus $|X|$ for the neighbors of x in X, minus e', minus the number of components that contain B-vertices, and minus the number of X-vertices that are adjacent by double edges to x. Hence, it holds that $c \geq 5|B| - |X| - e' - 2|B| = 3|B| - |X| - e'$. Together with $3|B| \geq 3|X| + e'$ this gives us $c \geq 2|X|$. Let Y be the set of these at least $2|X|$-many components in $\mathcal{C}$, which induce trees, do not intersect B and are incident with exactly one edge to x each.

Let A be a bipartite graph with vertex bipartition $V(A) = X\dot\cup Y$. Add an edge between $v \in X$ and $C \in Y$ if and only if there exists an edge between v and C in G. Let us fix any $C \in Y$ and show that it is adjacent to some $v \in X$ in A. Since C is a subgraph of R, and because we cannot apply Reduction Rule 8 or Reduction Rule 9, every vertex in C has degree at least 3 in G, including the leaves of C. Since every leaf of C is incident with at most one edge to x, and has at most one edge inside of C, it needs to be also adjacent to X, and thus C is adjacent to some vertex $v \in X$ in A. Together with the fact that $|Y| \geq 2|X|$, we can apply Lemma 22 and find nonempty sets $X' \subseteq X$ and $Y' \subseteq Y$, such that $N_A(Y') = X'$, and such that every subset $Z \subseteq X'$ has at least $2|Z|$ neighbors in Y'. Summing up, we have that every element C of Y' is a connected component of the graph $G\setminus(X'\cup\{x\})$, and C is adjacent to x through exactly one edge, C does not intersect B and induces a tree on G, and that every subset Z of X' is adjacent to at least $2|Z|$ components in Y'. Hence, we can apply Reduction Rule 12. □

Theorem 6. *The parameterized problem* FEEDBACK VERTEX SET[fvs] *admits a polynomial boundaried kernelization with at most $\mathcal{O}((|B|+k)^2)$ vertices.*

Proof. Assume by Lemma 3 that we are given a boundaried graph G_B such that $G-B$ is a forest. Note that each among Reduction Rules 8 to 12 decreases the value $n+s$, where n and s are the number of vertices and simple edges of G, respectively. This gives a bound linear on $|G|$ for the number of times we apply these reduction rules.

Let G'_B be the result after exhaustively applying Reduction Rules 8 to 11 and Lemma 23, and let $R' = V(G')\setminus B$. Observe, that there are less than $5|B|^2$ many edges between B and $|R'|$. Since we cannot apply Reduction Rule 8 and Reduction Rule 9, every vertex in R' has degree at least three in G'. Hence, each leaf in R', as well as each non-branching inner vertex of R', has at least one B-neighbor, which means that

there are less than $5|B|^2$-many leaves and non-branching vertices in R'. Further, the number of branching vertices in a forest is upper bounded by the number of leaves. Together, this yields a size bound of at most $\mathcal{O}(|B|^2)$ for R', and thus for the size of $V(G') = B \cup R'$. The output is thus G'_B with B being a modulator to forest, and $\Delta = 0$. □

4.4 Long Cycle[vc] and related problems

Our next positive results regarding polynomial boundaried kernelization are for problems that are not formulated in terms of vertex-deletion to some graph class $\mathcal{C}$, namely LONG CYCLE[vc], LONG PATH[vc], HAMILTONIAN CYCLE[vc], HAMILTONIAN PATH[vc], HAMILTONIAN CYCLE[$\#v, \deg(v) \neq 2$], HAMILTONIAN PATH[$\#v, \deg(v) \neq 2$]. These problems admit polynomial kernelization, as was shown by Bodlaender et al. [7]. Mainly, our argumentation will work very similarly to theirs, however, we are not able to easily reduce LONG PATH[vc] to LONG CYCLE[vc] by simply adding a global vertex which is adjacent to all original vertices, as we can only modify G_B, and thus such a vertex would not be adjacent to all vertices of H in the glued instance $G_B \oplus H_B$. Still, a very similar argumentation to the kernelization for LONG CYCLE[vc] is able to help us out.

We first introduce a result of Bodlaender et al. which will be our main tool.

Lemma 24 ([7], Theorem 2)**.** *Let $G = (X \dot\cup Y, E)$ be a bipartite graph. Let $M \subseteq E(G)$ be a maximum matching in G. Let $X_M \subseteq X$ be the set of vertices in X that are endpoints of an edge in M. Then, for each $Y' \subseteq Y$, if there exists a matching M' in G that saturates Y', then there exists a matching M'' in $G[X_M \cup Y]$ that saturates Y'.*

For G_B, construct the bipartite graph A, with one side of the vertex set of A being R, and the other consisting of pairs of distinct vertices in B. There is an edge between $v \in R$ and $\{p, q\} \subseteq B$, if and only if v is adjacent to both p and q. Let M_A be a maximum matching in A and let $J_A \subseteq R$ be the set of those vertices that are matched by M_A. If G contains cycles of size 4 with exactly two vertices in R, then fix one such cycle C'. If such C' exists, then let K contain the two R-vertices of C', and B_K the two B-vertices of C'. If no such C' exists, then let $K = B_K = \emptyset$.

Reduction Rule 13 (Analogue of [7, Reduction Rule 1])**.** *Delete all vertices in $R \setminus (J_A \cup K)$ from G_B.*

Proof of gluing safeness. Let G'_B be the resulting boundaried graph. Let H_B be any boundaried graph to be glued to G_B, resp., to G'_B. Since $G'_B \oplus H_B$ is a subgraph of $G_B \oplus H_B$, it clearly holds that $\mathrm{OPT}_{\mathrm{LC}}(G_B \oplus H_B) \geq \mathrm{OPT}_{\mathrm{LC}}(G'_B \oplus H_B)$. Let us also show that the reverse is true.

Let C be a cycle in $G_B \oplus H_B$. Clearly, whenever a vertex $v \in V(R)$ is visited by C, it holds that C visited some vertices in B before and after v. Let $v_1, \ldots, v_r$ be all the vertices of R contained in C, and let p_i, q_i be the predecessor and successor of v_i in C, respectively.

Let us first inspect the case that C has length 4 and $r = 2$, i.e., $C = (p_1, v_1, q_1, v_2, p_1)$. Since C witnesses the existence of a size-4 cycle with exactly two vertices in R, it holds true that exists a cycle C' of size 4, and since its vertices remain in G'_B, we know that C' exists in $G'_B \oplus H_B$.

Now we can assume that there are no $i, j \in [r]$ with $i \neq j$ and $\{p_i, q_i\} = \{p_j, q_j\}$. Let $W = \{\{p_i, q_i\} \mid i \in [r]\}$. Note that there exists a W-saturating matching in A, namely matching each $\{p_i, q_i\}$ with v_i. By Lemma 24, there also exists a W-saturating matching M' in $A[W \cup J_A]$.

For every $i \in [r]$, let v'_i denote the vertex to which $\{p_i, q_i\}$ is matched by M'. It is easy to see that we may replace each v_i on C by v'_i since v'_i is adjacent to p_i and q_i in G, obtaining a cycle C^* which intersects I only at J_A. As all pairs $\{p_i, q_i\}$ are different, no vertex v'_i is required twice. Hence C^* is also a cycle of $G'_B \oplus H_B$ of same size as C. □

Constructing A and computing M_A, as well as applying Reduction Rule 13 clearly can all be done in polynomial time. Moreover, J_A has size at most $|B|^2$, and thus after application of Reduction Rule 13, R contains at most $|B|^2 + 2 \in \mathcal{O}(|B|^2)$ vertices. Together with Lemma 3 we get the following result.

Theorem 7. *The parameterized problem* LONG CYCLE[vc] *admits a polynomial boundaried kernelization with at most $\mathcal{O}((|B| + k)^2)$ vertices.*

For LONG PATH[vc] we construct a different auxiliary bipartite graph, let it be denoted by D. Let the vertex set of D consist of R on one side and let the other side of $V(D)$ be the union of B and the set of pairs of distinct vertices in B. We compute a maximum matching M_D in D and denote by $J_D \subseteq I$ the set of those vertices that are matched by M_D.

Reduction Rule 14. *Delete all vertices in $R \setminus J_D$ from G_B.*

Proof of gluing safeness. Let G'_B be the resulting boundaried graph. Let H_B be any boundaried graph to be glued to G_B, resp., to G'_B. Since $G'_B \oplus H_B$ is a subgraph of $G_B \oplus H_B$, it clearly holds that $\mathrm{OPT}_{\mathrm{LP}}(G_B \oplus H_B) \geq \mathrm{OPT}_{\mathrm{LP}}(G'_B \oplus H_B)$. Let us also show that the reverse is true.

Let P be a path in $G_B \oplus H_B$. Let R' be the set of R-vertices visited by P. We denote the vertices of R' in the order they appear in P, i.e., $R' = \{v_1, \ldots, v_r\}$, and for every $i, j \in [r]$ with $i < j$ it holds that P visits v_i before v_j.

Assume that every R'-vertex v_i has a predecessor p_i and a successor q_i in P, both of which are obviously contained in B. Let $W = \{\{p_i, q_i\} \mid i \in [r]\}$. Note that there exists a W-saturating matching in D, namely matching each $\{p_i, q_i\}$ with v_i. By Lemma 24, there also exists a W-saturating matching M' in $D[W \cup J_D]$. For every $i \in [r]$, let v'_i denote the vertex to which $\{p_i, q_i\}$ is matched by M'. It is easy to see that we may replace each v_i on P by v'_i since v'_i is adjacent to p_i and q_i in G, obtaining a path P' which intersects I only at J_D. As all pairs $\{p_i, q_i\}$ are different, no vertex v'_i is required twice. Hence P' is also a path of $G'_B \oplus H_B$ of same size as P.

Now assume that P starts with v_1 and ends with v_r, i.e., v_1 has only a successor q_1 in P, and v_r only a predecessor p_r. Then let $W = \{q_1\} \cup \{p_r\} \cup \{\{p_i, q_i\} \mid i \in \{2, \ldots, r-1\}\}$. Again, W has a saturating matching in D, namely matching v_1 with q_1, and v_r with p_r, and for each $i \in \{2, \ldots, r-1\}$, matching $\{p_i, q_i\}$ with v_i. Using Lemma 24, we get a W-saturating matching M' in $D[W \cup J_D]$. Denote by v'_1 the vertex to which q_1 is matched by M', by v'_r the vertex to which p_r is matched by M', and for each $i \in \{2, \ldots, r-1\}$, denote by v'_i the vertex to which $\{p_i, q_i\}$ is matched by M'. Again, it is easy to see that we may replace each v_i on P by v'_i and obtain a path P', which intersects I only at J_D and is thus also a path of $G' \oplus H_B$.

There are two other cases: (i) P starts with v_1 but does not end with v_r; and (ii) P does not start with v_1 but ends with v_r. It is straightforward to handle these two cases in a similar manner as the first two cases were handled. □

Again, constructing D and computing M_D, as well as applying Reduction Rule 14 clearly can all be done in polynomial time. Moreover, J_D has size at most $|B|^2 + |B|$, and thus after application of Reduction Rule 14, R contains at most $|B|^2 + |B| \in \mathcal{O}(|B|^2)$ vertices. Together with Lemma 3 we get the following result.

Theorem 8. *The parameterized problem* LONG PATH[vc] *admits a polynomial boundaried kernelization with at most $\mathcal{O}((|B| + k)^2)$ vertices.*

Theorem 9. *The parameterized problems* HAMILTONIAN CYCLE[vc], HAMILTONIAN PATH[vc] *each admit a polynomial boundaried kernelization with at most $\mathcal{O}(|B| + k)$ vertices.*

Proof. Assume that $G - B$ is an independent set and apply Lemma 3 to lift this assumption and get a polynomial boundaried kernelization with $\mathcal{O}(|B| + k)$ vertices.

If $|V(G)| \leq 2|B| + 1$, we are done. Else for any H_B there is no Hamiltonian cycle and no Hamiltonian path on $G_B \oplus H_B$. Assume for contradiction that such a cycle or path exists, and denote it by P. In particular, P should visit all vertices in R. Let us be given the ordering of the R-vertices in which they are visited by P, i.e., let $R = \{v_1, \ldots, v_r\}$ with $r = |B|$. If P is a cycle, then simply choose any of the R-vertices as the first one. By definition, for each $i \in [r-1]$ there needs to be a path from v_i to v_{i+1}, and since R is an independent set, this path needs to visit at least one B-vertex. As the inner vertices of these paths need to be disjoint for any $i_1, i_2 \in [r-1]$ with $i_1 \neq i_2$, it follows that P needs to visit at least $|R| - 1$ different B-vertices. By assumption, it holds that $|R| > |B| + 1$, i.e., $|B| < |R| - 1$, and as a result, our assumed path/cycle P cannot exist. □

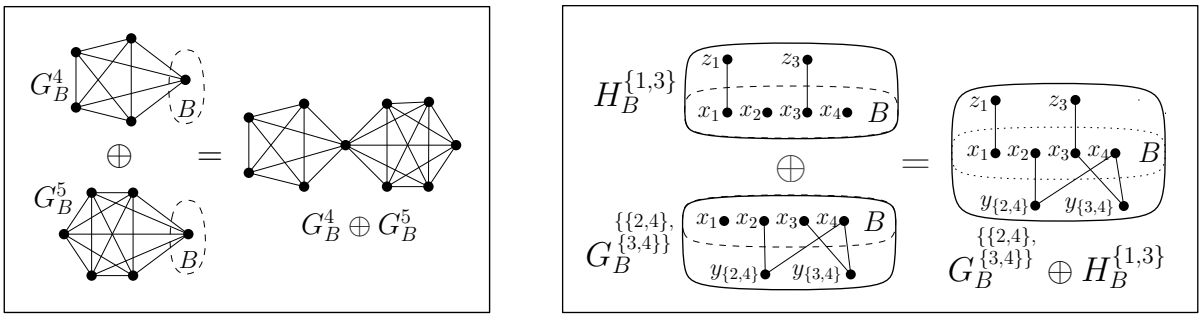

Figure 1: Examples of graphs defined in proofs for Lemmas 25 (left) and 31 (right).

Theorem 10. *Each among the parameterized problems* HAMILTONIAN CYCLE$[\#v, \deg(v) \neq 2]$ *and* HAMILTONIAN PATH$[\#v, \deg(v) \neq 2]$ *admits a polynomial boundaried kernelization with at most* $\mathcal{O}(|B|+k)$ *vertices.*

Proof. Let us be given a boundaried graph G_B and let X be the set of vertices in G with degree different than two. Let $R_X = V(G - (B \cup X))$. It is easy to see the gluing safeness of contracting any two adjacent vertices u, v in R_X to some vertex $w_{u,v}$: Computing a solution for $G_B \oplus H_B$ from one for $G'_B \oplus H_B$ can be done by replacing $w_{u,v}$ by u, v in the correct order; for the other direction we can assume without loss of generality, that v appears directly after u in the solution, which lets us replace u, v by $w_{u,v}$. Exhaustive application of this reduction rule leaves us with $B \cup X$ being a vertex cover. Hence apply the polynomial boundaried kernelization from Theorem 9 to obtain gluing-equivalent graph $\hat{G}_B$ and offset Δ. Further, it still holds that X contains all vertices in $\hat{G}$ with degree different than two. □

5 Lower bounds

Despite the relatively good behavior of the polynomial kernels used to obtain boundaried counterparts in Section 4, one cannot expect this to be the case for all problems. Among such, we show CLUSTER EDITING[ce] and TREE DELETION SET[tds] to not admit any boundaried kernelization, although polynomial kernels are known for CLUSTER EDITING$[k]$ and TREE DELETION SET$[k]$ [31, 13, 30]. Note that (polynomial) kernelization for $\Pi[k]$ implies such also for $\Pi[\Pi]$ with Π being a pure graph minimization problem: given (G, k, s), if $\Pi(G, s) \leq k$, then answer YES, else a (polynomial) bound on k also implies such on $\Pi(G, s)$.

Most of our lower bound results are obtained through excluding finite integer index, which was used as a sufficient condition for effective protrusion replacement earlier [6]. Furthermore, we also show a "too large" integer index in order to unconditionally rule out a polynomial boundaried kernelization for the problem DOMINATING SET[vc], which is consistent with Lemma 4 and an existing conditional exclusion of a polynomial kernel for the problem [17]. A similar result is obtained by Jansen and Wulms [37] who showed that VERTEX COVER and DOMINATING SET do not admit a single-exponential number of gluing-equivalence classes, even with the restriction to planar graphs with treewidth at most $|B| + \mathcal{O}(1)$. Together with Lemma 5 their result excludes polynomial boundaried kernelization for VERTEX COVER[tw + mod to planar] and DOMINATING SET[tw + mod to planar].

5.1 Cluster Editing

Lemma 25. CLUSTER EDITING *does not have finite integer index, even on the class of complete graphs.*

Proof. Choosing our boundary B to consist of exactly one vertex x, we show that the equivalence relation $\equiv_{\mathrm{CE},B}^{\mathcal{C}^{\mathtt{complete}}}$, from now on denoted simply $\equiv$ throughout this proof, has infinitely many equivalence classes, by giving an infinite set of boundaried graphs $G_B^i, i \in \mathbb{N}$ with G^i being complete, such that for any distinct $i, j \in \mathbb{N}$ it holds that $G_B^i \not\equiv G_B^j$. Namely, we define G_i as a complete graph on $i + 1$ vertices, with the

boundary vertex x being any one of the vertices. It is easy to see that these graphs are all non-isomorphic. The boundaried graph G^0_B consists solely of the vertex x.

For any $i \in \mathbb{N}$ it holds that $G^i_B \oplus G^0_B = G^i$, which is a clique on $i+1$ vertices, and hence the optimum CLUSTER EDITING solution for this graph has size zero. Similarly, we show that for any $i, j \in \mathbb{N}$ with $i > j$ it holds that $\mathrm{OPT}_{\mathrm{CE}}(G^i_B \oplus G^j_B) = j$. Without loss of generality we will assume that the vertex sets of G^i and G^j intersect at x only. Let $G = G^i_B \oplus G^j_B$ and denote $E(G)$ by E. Observe that G consists of two cliques, one of size $i+1$ and the other of size $j+1$, intersecting exactly at x only. Let $C^i = V(G^i - x)$ and $C^j = V(G^j - x)$. We can choose $S = E(x, C^j) = \{\{x, v\} \mid v \in C^j\}$ as a solution for G of size j. One easily sees, that $G \triangle S$ consists of two cliques, $C^i \cup \{x\}$ and C^j. Assume that there is some better solution S' of size less than j. It is clear that $E(x, C^j), E(x, C^i) \nsubseteq S'$, as otherwise we would have $|S'| \geq j$. Thus in $G \triangle S'$, vertex x is still adjacent to some vertices in both C^i and C^j. Let $v \in C^j$ be a vertex in C^j that remains adjacent to x in $G \triangle S'$. Similarly, let $U \subseteq C^i$ be the (maximal) set of all vertices in C^i that remain adjacent to x in $G \triangle S'$. As (u, b, v) with $u \in U$ forms an induced P_3 in G, it follows that we need the edge $\{u, v\}$ as well in S', to make it a triangle. By maximality of U, any vertex in $C^i \setminus U$ needs to be disconnected from x in $G \triangle S'$, and thus all edges in $E(x, C^i \setminus U)$ need to be in S'. Counting the number of edges already found to be in S', we find that S was actually smaller: $|S'| \geq |E(U, v)| + |E(x, C^i \setminus U)| \geq |C^i| = i > j$.

Now, let us be given some fixed $i, j \in \mathbb{N}$ with $i \neq j$, let h be a natural number larger than i and j. By previous argumentation, we know that $\mathrm{OPT}_{\mathrm{CE}}(G^i_B \oplus G^0_B) = \mathrm{OPT}_{\mathrm{CE}}(G^j_B \oplus G^0_B) = 0$, but at the same time, we know that $\mathrm{OPT}_{\mathrm{CE}}(G^i_B \oplus G^h_B) = i$, while $\mathrm{OPT}_{\mathrm{CE}}(G^j_B \oplus G^h_B) = j \neq i$. This way, there exists no constant Δ to witness gluing equality of G^i_B and G^j_B and we get $G^i_B \not\equiv G^j_B$. □

By Lemma 5 and the fact that graphs in $\mathcal{C}^{\texttt{complete}}$ have $\emptyset$ as a solution with value 0 for both CLUSTER EDITING and CLUSTER VERTEX DELETION, it follows from Lemma 25:

Theorem 11. *The parameterized problems* CLUSTER EDITING[cvd] *and* CLUSTER EDITING[ce] *do not admit boundaried kernelization.*

5.2 Maximum Cut

Lemma 26. MAXIMUM CUT *does not have finite integer index, even on boundaried graphs with $G - B$ being an independent set.*

Proof. We fix the boundary B to contain exactly two vertices, x and y, and define an infinite series of boundaried graphs G^i_B with $i \in \mathbb{N}$ such that $i \neq j$ leads to $G^i_B \not\equiv G^j_B$. Namely, G^i_B consists of x, y and an independent set of size i, in which all vertices are adjacent to both x and y. In order to show that these graphs are not gluing-equivalent with respect to MAXIMUM CUT, we also define the graph H^i_B, which consists of a complete bipartite graph with i-vertices on each side. In one of the sides, every vertex is adjacent to x, and in the other one, every vertex is adjacent to y.

Note that for any fixed $i \in \mathbb{N}$, the graph G^i is constructed in such a way, that its whole edge set is a cut of size $2i$ between the boundary and non-boundary vertices. This leads to $\mathrm{OPT}_{\mathrm{MC}}(G^i_B \oplus H^0_B) = 2i$, as $G^i_B \oplus H^0_B = G^i$.

On the other hand, for $i < j$ it holds that $\mathrm{OPT}_{\mathrm{MC}}(G^i_B \oplus H^j_B) = j^2 + 2j + i$: With $V \coloneqq V(G^i) \setminus B$ and $U \coloneqq N(x) \cap (V(H^j) \setminus B)$ and $W \coloneqq N(y) \cap (V(H^j) \setminus B)$ we get a corresponding solution by using the cut between $U \cup \{y\} \cup V$ and $W \cup \{x\}$ of size $|U| \cdot |W| + |U| + |W| + |V| = j^2 + 2j + i$. Let us argument why no better solution can be found. As we soon will show, to find a cut at least as large, we would still need U entirely in one half and W in the other, but then we for sure lose the edges between x and either U or W, hence we still get a solution of size at most $j^2 + 2j + i$. Say, the hypothetical solution is a cut between vertex sets X and Y. Let $U_X = U \cap X$ and define U_Y, W_X and W_Y analogically. Let $a = |U_X|$ and $b = |W_X|$. Since the total number of edges in $G \coloneqq G^i_B \oplus H^j_B$ is $2i + 2j + j^2$, and since edges between U_X and W_X (respectively, between U_Y and W_Y, of which there are $j - a$ and $j - b$) are not part of the cut, it holds that $|E_G(X, Y)| \leq j^2 + 2j + 2i - (ab + (j - a)(j - b))$. Let $f(a, b) = ab + (j - a)(j - b)$. In order to obtain $|E_G(X, Y)| \geq j^2 + 2j + i$, it thus must hold that $f(a, b) \leq i$. We do a case distinction: If $a = 0$ and $b = j$

or the other way around, i.e., it holds that U and W are completely contained in the opposing sides, we get $f(a,b) = 0$. If a and b are both zero or both t, then we have $f(a,b) = j^2 + 0 > i$. Else assume w.l.o.g. that it holds $1 \leq a \leq j-1$, which then implies that $ab \geq b$ and $(j-a)(j-b) \geq j-b$, hence $f(a,b) \geq b+j-b = j > i$.

To conclude this proof, assume we are given arbitrary, unequal $i, j, h \in \mathbb{N}$ and let h be larger than both i and j. Then h witnesses that G_B^i and G_B^j are not gluing equivalent with respect to MAXIMUM CUT, as we can see by previous argumentation, that $\mathrm{OPT}_{\mathrm{MC}}(G_B^i \oplus H_B^0) - \mathrm{OPT}_{\mathrm{MC}}(G_B^j \oplus H_B^0) = 2i - 2j$ is not equal to $\mathrm{OPT}_{\mathrm{MC}}(G_B^i \oplus H_B^h) - \mathrm{OPT}_{\mathrm{MC}}(G_B^j \oplus H_B^h) = i - j$. □

From Lemma 26 and Lemma 5 directly follows:

Theorem 12. *The parameterized problem* MAXIMUM CUT[vc] *does not admit boundaried kernelization.*

5.3 Tree Deletion Set

Lemma 27. TREE DELETION SET *does not have finite integer index, even on boundaried graphs with $G - B$ being an independent set.*

Proof. Choosing our boundary B to consist of exactly one vertex x, we show that the equivalence relation $\equiv_{\mathrm{TDS},B}^{\mathcal{C}_B^{\texttt{independent}}}$, from now on denoted simply $\equiv$ in this proof, has infinitely many equivalence classes. For this, we give an infinite set of boundaried graphs $G_B^i, i \in \mathbb{N}_{\geq 1}$ with $G_B^i \in \mathcal{C}_B^{\texttt{independent}}$, such that for any distinct $i, j \in \mathbb{N}$ it holds that $G_B^i \not\equiv G_B^j$. We define G^i to be an i-star with x as the center, i.e., G^i is a graph containing the boundary vertex x and i other vertices $v_1, \ldots, v_i$, which are all adjacent to x. In order to show non-equivalence of G_B^i and G_B^j, we define additional boundaried graphs $H_B^h, h \in \mathbb{N}_{\geq 1}$, each of which consists of vertices r, a_n, b_n with $n \in [h]$ and the boundary vertex x. For each $n \in [h]$ the vertices a_n and b_n are adjacent to each other and x; furthermore, a_n is adjacent to r.

With those structures defined, fix any $i, j \in \mathbb{N}_{\geq 1}$ with $i < j$, and let $h = j + 1$. First, note that $\mathrm{OPT}_{\mathrm{TDS}}(G_B^i \oplus H_B^1) = \mathrm{OPT}_{\mathrm{TDS}}(G_B^j \oplus H_B^1) = 1$, since both resulting graphs contain exactly one cycle which is induced by $\{a_1, b_1, x\}$, which can be hit by removing b_1, while preserving connectivity. On the other hand, we show that $\mathrm{OPT}_{\mathrm{TDS}}(G_B^i \oplus H_B^h) \leq i + 1$ while $\mathrm{OPT}_{\mathrm{TDS}}(G_B^j \oplus H_B^h) \geq h = j + 1 > i + 1$, which then proves our point, as we do not have the same difference in optimum values after gluing with H_B^1 and H_B^h.

Note that each $F^i := G_B^i \oplus H_B^h$ and $F^j := G_B^j \oplus H_B^h$ contain h-many cycles, which pairwise all intersect exactly at x. This constellation is called an x-flower of order h, and it is easy to see, that in order to hit those h cycles with less than h vertices, one needs to take x. Both $F^i - x$ and $F^j - x$ are disconnected, but at the same time $T := \{r\} \cup \{a_n, b_n \mid n \in [h]\}$ induces a tree. Since this tree contains more than h vertices, we check whether removing all the other components leads to a solution smaller than h. In case of $F^i - x$, it is exactly the vertices $v_1, \ldots, v_i$ which are not contained in T, and whose removal gives us a feasible solution of size $i + 1$. However, in $F^j - x$, outside of T there are j-many vertices $v_1, \ldots, v_j$. Removing those would give us size $j + 1 = h$. Thus, F^j does not have a solution which is smaller than h. □

Lemma 28. TREE DELETION SET *does not have finite integer index on $\mathcal{C}_B^{\texttt{tree}}$.*

Proof. This proof works similarly to the proof for Lemma 27. We again choose $B = \{x\}$ and for $h \in \mathbb{N}_{\geq 1}$ we define the boundaried graph H_B^h exactly the same. The boundaried graph $G_B^i, i \in \mathbb{N}_{\geq 2}$ is defined nearly the same, but there is an extra vertex $y \neq x$, which is adjacent to all $v_1, \ldots, v_i$.

Let $i, j \in \mathbb{N}_{\geq 2}$ with $i < j$, and we emphasize that this time it holds that $i, j \geq 2$, but we still get an infinite amount of non-equivalent boundaried graphs. First, it holds that $\mathrm{OPT}(G_B^i \oplus H_B^1) = \mathrm{OPT}(G_B^j \oplus H_B^1) = 2$. This time, we have multiple cycles, which each contain at least either y or b_1. Removing these two vertices does not interrupt connectedness of the rest, so in both cases the optimum is at most 2. However, we cannot find any better solution: in search of one we would need to take x, as it is the only vertex contained in all cycles. This way we get multiple components, at least one of which then also needs to be contained in the solution. Hence such a solution has at least 2 vertices.

Now with $h = j + 2$, we show that $\mathrm{OPT}(G_B^i \oplus H_B^h) \leq i + 2$, while $\mathrm{OPT}(G_B^j \oplus H_B^h) \geq h = j + 2 > i + 2$. Let $F^i = G_B^i \oplus H_B^h$ and $F^j = G_B^j \oplus H_B^h$. For F^i it is sufficient to remove all vertices contained in G^i, i.e.,

we remove $x, y, v_1, \ldots, v_i$, which are $i+2$ vertices. This way we are left with the tree $H^h - x$. In order to find a solution for F^j of size less than h, we need to remove x, as the graph contains an x-flower of order at least h. However, $F^j - x$ contains two components, one with $2h+1$ vertices and the other with $j+1$. Whichever component we would choose to remove, we would get size at least $j+2$. This way, F^j does not have a solution of size less than h.

Altogether, we get $\mathrm{OPT}(G^j_B \oplus H^1_B) - \mathrm{OPT}(G^i_B \oplus H^1_B) = 0$, which is clearly not equal to $\mathrm{OPT}(G^j_B \oplus H^h_B) - \mathrm{OPT}(G^i_B \oplus H^h_B) \geq 1$. Hence it holds that $G^i_B \not\equiv G^j_B$. □

By Lemma 5 in combination with Lemmas 27 and 28 we get the following theorem.

Theorem 13. *The parameterized problems* TREE DELETION SET[vc] *and* TREE DELETION SET[tds] *do not admit boundaried kernelization.*

5.4 Long Cycle and Long Path

Lemma 29. LONG CYCLE *does not have finite integer index, even on the class of B-boundaried graphs where each $v \in V(G) \setminus B$ has degree two.*

Proof. Fix $B = \{x_1, x_2, x_3, x_4\}$ and define for each $i \in \mathbb{N}$ the graph G^i_B with additional vertices a and $b_1, \ldots, b_{i+1}$ and add edges such that (x_1, a, x_2) forms a path of length 2 and $(x_3, b_1, \ldots, b_{i+1}, x_4)$ forms a path of length $i+2$. Further define the following two graphs: L_B with additional vertex l such that (x_1, l, x_2) forms a path of length 2; R_B with additional vertex r such that (x_3, r, x_4) forms a path of length 2. It is easy to see that for any distinct $i, j \in \mathbb{N}$ it holds that $\mathrm{OPT}(G^i_B \oplus L_B) = 4 = \mathrm{OPT}(G^j_B \oplus L_B)$, while $\mathrm{OPT}(G^i_B \oplus R_B) = i + 4 \neq j + 4 = \mathrm{OPT}(G^j_B \oplus R_B)$, implying that $G^i_B \not\equiv G^j_B$. □

Lemma 30. LONG PATH *does not have finite integer index, even on the class of B-boundaried graphs where each $v \in V(G) \setminus B$ has degree two.*

Proof. Fix $B = \{x_1, x_2, x_3\}$ and define for each $i \in \mathbb{N}$ the graph G^i_B with additional vertices $b_1, \ldots, b_{i+1}$ and edges such that $(x_2, b_1, \ldots, b_{i+1}, x_3)$ forms a path of length $i+2$. Further let H^i_B have additional vertices $c_1, \ldots, c_i$ such that $(x_1, c_1, \ldots, c_i)$ forms a path of length i. Now given distinct $i, j \in \mathbb{N}$ (and assume without loss of generality that $i > j$) and $h = i + 3$ it is easy to see that $\mathrm{OPT}(G^i_B \oplus H^h_B) = h = \mathrm{OPT}(G^j_B \oplus H^h_B)$, while $\mathrm{OPT}(G^i_B \oplus H^0_B) = i + 2 > j + 2 = \mathrm{OPT}(G^j_B \oplus H^0_B)$, implying that $G^i_B \not\equiv G^j_B$. □

By Lemma 5 and Lemmas 29 and 30 we get:

Theorem 14. LONG CYCLE$[\#v, \deg(v) \neq 2]$ *and* LONG PATH$[\#v, \deg(v) \neq 2]$ *do not admit polynomial boundaried kernelization.*

5.5 Dominating Set

One especially interesting case in our series of lower bounds for boundaried kernelization is DOMINATING SET[vc], as DOMINATING SET is known to have finite integer index (see [6]), but by showing, that it does not have single-exponential integer index, we can still rule out any polynomial boundaried kernelization for DOMINATING SET[vc]. This time, we cannot choose B to consist just of one or two vertices, and instead need to work with B having arbitrary size.

Lemma 31. DOMINATING SET *does not have single-exponential integer index, even on $\mathcal{C}^{independent}_B$.*

Proof. Without loss of generality, let $B = \{x_1, \ldots, x_q\}$ for some arbitrary, but fixed $q \in \mathbb{N}$. We give a set $\mathcal{Q}$ of $2^{\binom{q}{\lfloor q/2 \rfloor}}$-many pairwise non-equivalent (with respect to $\equiv$) boundaried graphs. It holds that $t \coloneqq \binom{q}{\lfloor q/2 \rfloor} \geq (\frac{q}{\lfloor q/2 \rfloor})^{\lfloor q/2 \rfloor} \geq 2^{\lfloor q/2 \rfloor} \in \omega(|B|^c)$ for any constant c. Hence, $|\mathcal{Q}|$ is not single-exponential. Let $\mathcal{D}$ be the family of all size-$\lfloor q/2 \rfloor$ subsets of $[q] = \{1, \ldots, q\}$. For any $\mathcal{I} \subseteq \mathcal{D}$ let $G^{\mathcal{I}}_B$ be a B-boundaried graph with additional vertices y_I for every $I \in \mathcal{I}$, each of which is adjacent to vertices x_i with $i \in I$. Since the vertices

y_I have no edges between each other, it clearly holds that $G_B^{\mathcal{I}} \in \mathcal{C}_B^{\texttt{independent}}$. We set $\mathcal{Q} \coloneqq \{G_B^{\mathcal{I}} \mid \mathcal{I} \subseteq \mathcal{D}\}$. Observe that $|\mathcal{Q}| = 2^t$, so it remains to show that different $G_B^{\mathcal{I}}, G_B^{\mathcal{J}} \in \mathcal{Q}$ are not gluing-equivalent. Therefore, we define for each $P \subseteq [q]$ a B-boundaried graph H_B^P with additional vertices z_i for each $i \in P$, which is adjacent only to x_i. Observe that after gluing any $G_B^{\mathcal{I}}$ to H_B^P, there is a minimum dominating set containing x_i for every $i \in P$, as it is adjacent to a vertex of degree one, which needs to be dominated anyway.

Let $\mathcal{I}$ and $\mathcal{J}$ be two different subsets of $\mathcal{D}$. We show that $G_B^{\mathcal{I}} \not\equiv G_B^{\mathcal{J}}$. First, note that both $G_B^{\mathcal{I}} \oplus H_B^{[q]}$ and $G_B^{\mathcal{J}} \oplus H_B^{[q]}$ have B as an optimum solution: By previous argumentation exists an optimum solution containing all x_i with $i \in [q]$, and all $y_I, I \in \mathcal{I}$ are adjacent to some vertices in B, thus all vertices are dominated by B. This way we get $\mathrm{OPT}(G_B^{\mathcal{J}} \oplus H_B^{[q]}) - \mathrm{OPT}(G_B^{\mathcal{I}} \oplus H_B^{[q]}) = 0$.

As $\mathcal{I} \neq \mathcal{J}$, we can assume without loss of generality, that exists some $I \in \mathcal{I} \setminus \mathcal{J}$. Consider the two graphs $F^{\mathcal{I}} = G_B^{\mathcal{I}} \oplus H_B^{[q]\setminus I}$ and $F^{\mathcal{J}} = G_B^{\mathcal{J}} \oplus H_B^{[q]\setminus I}$. We show that $\mathrm{OPT}(F^{\mathcal{J}}) - \mathrm{OPT}(F^{\mathcal{I}}) \neq 0$, which proves that $G_B^{\mathcal{I}}$ and $G_B^{\mathcal{J}}$ are not gluing equivalent. For this, first observe that $\mathrm{OPT}(F^{\mathcal{I}}) \leq \lceil q/2 \rceil + 1$, since the set $\{y_I\} \cup \{x_h \mid h \in [q] \setminus I\}$ is a feasible solution of that size: for each $h \in [q] \setminus I$ it holds that x_h dominates x_h and z_h; each y_J with $J \in \mathcal{I} \setminus \{I\}$ is also dominated by some x_h with $h \in [q] \setminus I$, and y_I together with all x_r with $r \in I$ are dominated by y_I. However, it holds that $\mathrm{OPT}(F^{\mathcal{J}}) > \lceil q/2 \rceil + 1$. By earlier argumentation we know it is safe to assume that for each $h \in [q] \setminus I$ the vertex x_h is in the optimum solution. As a next step, we would need to dominate all x_r with $r \in I$ using only one vertex. However, by construction no vertex in $F^{\mathcal{J}}$ is adjacent to all such x_r. □

Using Lemmas 5 and 31 we get the following theorem.

Theorem 15. *The parameterized problem* DOMINATING SET[vc] *does not admit polynomial boundaried kernelization.*

6 Improved kernelization for VC[mod to constant td]

By applying the idea of boundaried kernelization we give an improved kernel for VERTEX COVER[mod to td $\leq d$]. Bougeret and Sau [12] gave the first polynomial kernel for this problem on general graphs, which was then improved and generalized by Hols et al. [32]. The kernelization works in a recursive manner: If $d = 1$, then apply the vertex-linear kernel for VERTEX COVER[vc]. Else, given a graph G and modulator X to constant treedepth $d > 1$, reduce the number of connected components in $G - X$ to at most $\mathcal{O}(|X|^{2^{d-2}+1})$, and put from each remaining component the root vertex of its treedepth decomposition to the modulator, reducing the task to computing a kernel for VERTEX COVER[mod to td $\leq d-1$]. In total, this yields a kernel with $\mathcal{O}(|X|^{2^{\Theta(d^2)}})$ vertices. An issue with this approach is that the roots of all components are moved to the modulator simultaneously, leading to an unnecessarily large blow-up of the number of components in each recursive step. Instead, we take a depth-first approach: At each recursive step, after having reduced the number of components, we partition the set of remaining components into groups of size at most $|X|$, and recursively process each group individually as a boundaried graph, with X and the roots of the respective components forming the boundary and modulator to treedepth $d-1$. This way, we achieve a kernel with $\mathcal{O}(|X|^{2^{d-1}})$ vertices, which comes much closer to the best known lower bound of $\mathcal{O}(|X|^{2^{d-2}+1})$ vertices [32, Theorem 1.1]. The correctness of this approach follows from Theorem 4 and Lemma 33, a *gluing*-equivalence restatement for bounding the number of connected components in $G - X$. We remind that the notion of chunks, blocking sets etc., was defined in Section 4.1.

Let A be a bipartite graph, whose vertex set consists of the connected components of $G - B$ on one side, and of the chunks of B on the other side. Add an edge between component F and chunk Z in A, if and only if $N_F(Z)$ is a blocking set of F, i.e., if it holds that $\textsc{conf}_F(Z) > 0$. Observe that by assumption of Lemma 33, we can compute $\textsc{conf}_F(Z)$ in polynomial time. Using the Hopcroft-Karp algorithm for maximum matching, find a set $\mathcal{X}' \subseteq \mathcal{X}$ such that $|N_A(\mathcal{X}')| < |\mathcal{X}'|$ and a matching M in A that saturates $\hat{\mathcal{X}} = \mathcal{X} \setminus \mathcal{X}'$. Note that if there exists an $\mathcal{X}$-saturating matching for A, then it holds that $\mathcal{X}' = \emptyset$ and $\hat{\mathcal{X}} = \mathcal{X}$. Denote by $\mathcal{F}_D$ the set of connected components of $G - B$ that are neither in the neighborhood of $\mathcal{X}'$, nor contained in a matching edge of M.

Lemma 32 (Analogue of [32, Lemma 3.4]). *For every H_B there exists an optimum vertex cover S of $G_B \oplus H_B$ with $S \cap Z \neq \emptyset$ for all $Z \in \hat{\mathcal{X}}$.*

Proof. Fix any boundaried graph H_B to be glued to G_B. We will go along the proof of Lemma 3.4 in the paper of Hols et al.

Let S be an optimum vertex cover of $G_B \oplus H_B$. If $S \cap Z \neq \emptyset$ for all $Z \in \hat{\mathcal{X}}$, then we are done. Thus, assume that there exists at least one set $Z \in \hat{X}$ such that $S \cap Z = \emptyset$. Let $\tilde{\mathcal{X}} = \{Z \in \hat{\mathcal{X}} \mid S \cap Z = \emptyset\}$ be the set of chunks with $S \cap Z = \emptyset$, and let $\tilde{\mathcal{F}} = \{H \in \mathcal{F} \mid \exists Z \in \tilde{\mathcal{X}} : \{Z, H\} \in M\}$ be the set of connected components of $G - B$, that are matched to a vertex in $\tilde{X}$ by M. Observe that $|\tilde{X}| = |\tilde{\mathcal{F}}|$. Hols et al. have shown that for every $F \in \tilde{\mathcal{F}}$ it holds that $|V(F) \cap S| > \mathrm{OPT}(F)$.

Now we construct an optimum vertex cover S' of $G_B \oplus H_B$ which fulfills the properties of the lemma. First, we add from each chunk $Z \in \tilde{\mathcal{X}}$ one arbitrary vertex to the set S. We denote the resulting set $\hat{S}$. It holds that $|\hat{S}| \leq |S| + |\tilde{X}|$. Furthermore, Hols et al. have shown that for every $F \in \tilde{\mathcal{F}}$ there exists an optimum vertex cover S_F of F which contains the set $Y_F = N(B \setminus \hat{S}) \cap V(F)$. This is exactly what we do in our next step: we replace for every connected component $F \in \tilde{\mathcal{F}}$ the set $S \cap V(F)$ in $\hat{S}$ by the optimum vertex cover S_F in F, that contains the set Y_F. Denote the resulting set by S'. As we have seen earlier, that for S and every $F \in \tilde{\mathcal{F}}$, the set $V(F) \cap S$ was larger than an optimum vertex cover for F, and we have replaced this part by an optimum vertex cover for F, it holds that $|S'| \leq |\hat{S}| - |\tilde{\mathcal{F}}| \leq |S| + |\tilde{\mathcal{X}}| - |\tilde{\mathcal{F}}| = |S|$.

It remains to show that S' is a vertex cover of $G_B \oplus H_B$. Since S' contains all vertices from $S \cap V(H)$, we know that every edge inside of H is still covered by S'. Every edge inside of $(G_B \oplus H_B)[B]$ is covered by S', since $S' \cap B$ is a superset of $S \cap B$. And at last, for every component $F \in \tilde{\mathcal{F}}$, the set S' contains all vertices of S_H, which covers all edges inside of F, and contains the neighborhood of all B-vertices that are not contained in S'. □

Reduction Rule 15 (Analogue of [32, Reduction Rule 3.3]). *Delete all connected components in $\mathcal{F}_D$ from G and increase Δ by $\mathrm{OPT}(\mathcal{F}_D)$.*

Proof of gluing safeness. Fix any boundaried graph H_B to glue with G_B. Let G'_B be the reduced instance, i.e., $G' = G - \mathcal{F}_D$. It obviously holds that $\mathrm{OPT}(G'_B \oplus H_B) \leq \mathrm{OPT}(G_B \oplus H_B) - \mathrm{OPT}(\mathcal{F}_D)$. So let us show, that the other direction also holds, i.e., that $\mathrm{OPT}(G_B \oplus H_B) \leq \mathrm{OPT}(G'_B \oplus H_B) + \mathrm{OPT}(\mathcal{F}_D)$. Note that with the same choice of $\hat{\mathcal{X}}$ we can also use Lemma 32 on G'_B. So let S' be a minimum vertex cover for $G'_B \oplus H_B$ with $S' \cap Z \neq \emptyset$ for all $Z \in \hat{\mathcal{X}}$. Note that in A, every connected component $F \in \mathcal{F}_D$ is only adjacent to vertices in $\hat{\mathcal{X}}$. Since every set $Z \in \hat{\mathcal{X}}$ has a nonempty intersection with the set S', it holds that there exists an optimum vertex cover S_F of F which contains the set $N_G(B \setminus S') \cap V(F)$ (similar to how it is the case for every $F \in \tilde{\mathcal{F}}$ in the proof of Lemma 32). Let S be the set that results from adding for each component $F \in \mathcal{F}_D$ the optimum vertex cover S_F to the set S'. By construction of S, it is a superset of S' and thus covers all edges in $G'_B \oplus H_B$. Additionally, all the new edges inside of any component $F \in \mathcal{F}_D$ and between F and $B \setminus S$ are covered by $S_F \subseteq S$. Thus, S is a vertex cover of $G_B \oplus H_B$ of size $|S'| + \mathrm{OPT}(\mathcal{F}_D)$. □

As Reduction Rule 15 works exactly the same as [32, Reduction Rule 3.3], we also refer to their proof (see [32, Lemma 3.10]), that Reduction Rule 15 can be applied in polynomial time. Similarly, we refer to Hols et al. for the bound of at most $\mathcal{O}(|B|^{bc})$ connected components after having applied Reduction Rule 15 ([32, Theorem 3.8]).

Lemma 33 (Analogue of [32, Theorem 1.3]). *Let $\mathcal{C}$ be a hereditary graph class with minimal blocking set size b on which* VERTEX COVER *can be solved in polynomial time. There is a polynomial-time algorithm that given boundaried graph $G_B \in \mathcal{C}_B$ returns a gluing-equivalent graph $G'_B \in \mathcal{C}_B$ with the respective offset Δ, such that $G' - B$ has at most $\mathcal{O}(|B|^b)$ connected components.*

Theorem 16. *For any constant $d \in \mathbb{N}_{\geq 1}$, the parameterized problem* VERTEX COVER[mod to td $\leq d$] *admits a polynomial boundaried kernelization with $\mathcal{O}((|B| + k)^{2^{d-1}})$ vertices.*

Proof. By Lemma 3, we assume that $G - B$ has treedepth d and show a boundaried kernelization with $\mathcal{O}(|B|^{2^{d-1}})$ vertices for this case. We do our proof by induction over d. In the base case it holds that $d = 1$,

which means that $G - B$ is an independent set. In this case we can use our boundaried kernelization for VERTEX COVER[vc] from Section 4.1 of vertex-size $\mathcal{O}(|B|)$.

Now let $d > 1$ and assume that VERTEX COVER[mod to td $\leq d-1$] with B being also the modulator, has a boundaried kernelization of vertex-size $\mathcal{O}(|B|^{2^{d-2}})$. We use Lemma 33 and the fact that $\mathcal{C}^{\mathrm{td}\leq d}$ has minimal blocking set size at most $2^{d-2}+1$ (see [32, Theorem 1.4]) in order to find a gluing-equivalent boundaried graph G'_B with $\mathcal{O}(|B|^{2^{d-2}+1})$ connected components. We partition these components into $t \in \mathcal{O}(|B|^{2^{d-2}+1})/|B|) = \mathcal{O}(|B|^{2^{d-2}})$ sets $Z_1, \ldots, Z_t$ by packing into each Z_i, $i \in [t]$ all the vertices of at most $|B|$ components. Let $R_1, \ldots, R_t$ be the sets of roots of the treedepth decompositions of $G^1 := G'[Z_1], \ldots, G^t := G'[Z_t]$. Note that for each $i \in [t]$ it holds that $|R_i| \leq |B|$ and that $G' = G^1_B \oplus \cdots \oplus G^t_B$. For each $i \in [t]$ it holds that $\mathrm{td}(G^i - (B \cup R_i)) \leq d-1$, and thus by induction hypothesis, we can compute in polynomial time a graph $\hat{G}^i$ of vertex size $\mathcal{O}((|B|+|R_i|)^{2^{d-2}}) = \mathcal{O}(|B|^{2^{d-2}})$ and constant Δ_i with $\mathrm{OPT}_{\mathrm{VC}}(G^i_{B\cup R_i} \oplus H_{B\cup R_i}) = \mathrm{OPT}_{\mathrm{VC}}(\hat{G}^i_{B\cup R_i} \oplus H_{B\cup R_i}) + \Delta_i$ for any $H_{B\cup R_i}$, and thus by Lemma 1 also $G^i_B \equiv_{\mathrm{VC},B} \hat{G}^i_B$ with same offset. By repeated use of Lemma 2, it holds that $G'_B \equiv_{\mathrm{VC},B} \hat{G}_B$ with $\hat{G} = \hat{G}^1_B \oplus \cdots \oplus \hat{G}^t_B$ and offset $\sum_{i=1}^t \Delta_i$. Thus, we can output the boundaried graph $\hat{G}_B$ which has vertex-size $t \cdot \mathcal{O}(|B|^{2^{d-2}}) = \mathcal{O}(|B|^{2^{d-1}})$. □

By Lemma 4 this also gives the regular polynomial kernel.

Theorem 3. *For any constant $d \in \mathbb{N}_{\geq 1}$, the parameterized problem* VERTEX COVER[mod to td $\leq d$] *admits a polynomial kernelization with $\mathcal{O}(k^{2^{d-1}})$ vertices.*

7 Conclusion

Boundaried kernelization is a model for efficient local preprocessing, inspired by protrusions and protrusion replacement used in meta kernelization. It allows for provably effective preprocessing when only local parts of the input exhibit useful structure, while this need not be true for the entire input. We have showed several polynomial-sized boundaried kernelizations as well as a number of unconditional lower bounds ruling out such preprocessing for other problems. This is unlike the typical use of protrusions where one can appeal to finite integer index to obtain a "small" replacement, which will usually be of size exponential in treewidth and boundary size (but still constant). That being said, finite (integer) index is still a prerequisite for having a boundaried kernelization of any size. Similarly, a (polynomial) kernelization is required for a problem to admit a (polynomial) boundaried kernelization.

While the present definition of boundaried kernelization is restricted to pure graph problems (parameterized by some graph optimization problem), there is no reason not to pursue such local kernelization also in more general settings. Rather, it was appealing to keep the notation (somewhat) concise by relying on the robust setting of boundaried graphs and the established notion of finite (integer) index. Moreover, even in the restricted setting of pure graph problems, we were able to showcase a number of different behaviors of target problems. Last but not least, this allowed us to define what is a boundaried kernelization for a given (parameterized) problem, rather than having to first define a specific boundaried version of each problem.

That being said, to go beyond pure graph problems necessitates clean definitions for what a local part of an instance should be. This is probably best formalized by having a problem-specific gluing operator on the level of entire instances. (As a simple example, consider gluing two boundaried instances STEINER TREE, where we should glue the graphs but also take the union of the respective terminal sets.) Ideally, such an operator has some form of neutral element/instance so that the connection to (regular) kernelization can be retained. As future work, we aim to extend boundaried kernelization (at least) to problems on general finite structures, which should be sufficient for a large(r) range of target problems while still enabling a generic definition without problem-specific parts.

Further directions of future work are, obviously, to research further what problems with polynomial kernelization also admit polynomial boundaried kernelizations (for pure graph problems and beyond), and to establish unconditional lower bounds where this is infeasible. It is also interesting to perhaps identify some general properties of problems that admit a polynomial kernelization but no polynomial boundaried kernelization, like, e.g., TREE DELETION SET.

Finally, we showed that our polynomial boundaried kernelization for VERTEX COVER[vc] can be used to get a polynomial kernelization for VERTEX COVER[mod to td $\leq d$], and in fact got a significant improvement over the previous kernelization. Can this be further improved to match the lower bound via maximum size of minimal blocking sets [32]? Can one find more applications of polynomial boundaried kernelization for obtaining polynomial kernelizations for more complex parameterizations?

References

[1] F. N. Abu-Khzam, M. R. Fellows, M. A. Langston, and W. H. Suters. Crown structures for vertex cover kernelization. *Theory Comput. Syst.*, 41(3):411–430, 2007.

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

[3] S. Assadi, S. Khanna, Y. Li, and V. Tannen. Dynamic sketching for graph optimization problems with applications to cut-preserving sketches. In P. Harsha and G. Ramalingam, editors, *35th IARCS Annual Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2015, December 16-18, 2015, Bangalore, India*, volume 45 of *LIPIcs*, pages 52–68. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.

[4] S. Bessy, M. Bougeret, D. M. Thilikos, and S. Wiederrecht. Kernelization for graph packing problems via rainbow matching. In N. Bansal and V. Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3654–3663. SIAM, 2023.

[5] S. Bhyravarapu, S. Jana, S. Saurabh, and R. Sharma. Difference determines the degree: Structural kernelizations of component order connectivity. In N. Misra and M. Wahlström, editors, *18th International Symposium on Parameterized and Exact Computation, IPEC 2023, September 6-8, 2023, Amsterdam, The Netherlands*, volume 285 of *LIPIcs*, pages 5:1–5:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[6] H. L. Bodlaender, F. V. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh, and D. M. Thilikos. (Meta) kernelization. *J. ACM*, 63(5):44:1–44:69, 2016.

[7] H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Kernel bounds for path and cycle problems. *Theor. Comput. Sci.*, 511:117–136, 2013.

[8] H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Kernelization lower bounds by cross-composition. *SIAM J. Discret. Math.*, 28(1):277–305, 2014.

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

[10] H. L. Bodlaender and T. C. van Dijk. A cubic kernel for feedback vertex set and loop cutset. *Theory Comput. Syst.*, 46(3):566–597, 2010.

[11] M. Bougeret, B. M. P. Jansen, and I. Sau. Kernelization dichotomies for hitting subgraphs under structural parameterizations. In K. Bringmann, M. Grohe, G. Puppis, and O. Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, July 8-12, 2024, Tallinn, Estonia*, volume 297 of *LIPIcs*, pages 33:1–33:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[12] M. Bougeret and I. Sau. How much does a treedepth modulator help to obtain polynomial kernels beyond sparse graphs? *Algorithmica*, 81(10):4043–4068, 2019.

[13] Y. Cao and J. Chen. Cluster editing: Kernelization based on edge cuts. *Algorithmica*, 64(1):152–169, 2012.

[14] J. Chen, I. A. Kanj, and W. Jia. Vertex cover: Further observations and further improvements. *J. Algorithms*, 41(2):280–301, 2001.

[15] H. Dell and D. van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM*, 61(4):23:1–23:27, 2014.

[16] R. Diestel. *Graph Theory, 5th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2017.

[17] M. Dom, D. Lokshtanov, and S. Saurabh. Incompressibility through colors and IDs. In S. Albers, A. Marchetti-Spaccamela, Y. Matias, S. E. Nikoletseas, and W. Thomas, editors, *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I*, volume 5555 of *Lecture Notes in Computer Science*, pages 378–389. Springer, 2009.

[18] H. Donkers and B. M. P. Jansen. A turing kernelization dichotomy for structural parameterizations of f-minor-free deletion. *J. Comput. Syst. Sci.*, 119:164–182, 2021.

[19] M. Dumas and A. Perez. An improved kernelization algorithm for trivially perfect editing. In N. Misra and M. Wahlström, editors, *18th International Symposium on Parameterized and Exact Computation, IPEC 2023, September 6-8, 2023, Amsterdam, The Netherlands*, volume 285 of *LIPIcs*, pages 15:1–15:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[20] S. Fafianie, E. C. Hols, S. Kratsch, and V. A. Quyen. Preprocessing under uncertainty: Matroid intersection. In P. Faliszewski, A. Muscholl, and R. Niedermeier, editors, *41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland*, volume 58 of *LIPIcs*, pages 35:1–35:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[21] S. Fafianie, S. Kratsch, and V. A. Quyen. Preprocessing under uncertainty. In N. Ollinger and H. Vollmer, editors, *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, volume 47 of *LIPIcs*, pages 33:1–33:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[22] M. R. Fellows and M. A. Langston. An analogue of the myhill-nerode theorem and its use in computing finite-basis characterizations (extended abstract). In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*, pages 520–525. IEEE Computer Society, 1989.

[23] M. R. Fellows, D. Lokshtanov, N. Misra, M. Mnich, F. A. Rosamond, and S. Saurabh. The complexity ecology of parameters: An illustration using bounded max leaf number. *Theory Comput. Syst.*, 45(4):822–848, 2009.

[24] F. V. Fomin, P. A. Golovach, T. Inamdar, S. Saurabh, and M. Zehavi. Kernelization for spreading points. In I. L. Gørtz, M. Farach-Colton, S. J. Puglisi, and G. Herman, editors, *31st Annual European Symposium on Algorithms, ESA 2023, September 4-6, 2023, Amsterdam, The Netherlands*, volume 274 of *LIPIcs*, pages 48:1–48:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[25] F. V. Fomin, T. Le, D. Lokshtanov, S. Saurabh, S. Thomassé, and M. Zehavi. Lossy kernelization for (implicit) hitting set problems. In I. L. Gørtz, M. Farach-Colton, S. J. Puglisi, and G. Herman, editors, *31st Annual European Symposium on Algorithms, ESA 2023, September 4-6, 2023, Amsterdam, The Netherlands*, volume 274 of *LIPIcs*, pages 49:1–49:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[26] F. V. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos. Kernels for (connected) dominating set on graphs with excluded topological minors. *ACM Trans. Algorithms*, 14(1):6:1–6:31, 2018.

[27] F. V. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos. Bidimensionality and kernels. *SIAM J. Comput.*, 49(6):1397–1422, 2020.

[28] F. V. Fomin, D. Lokshtanov, S. Saurabh, and M. Zehavi. *Kernelization: Theory of Parameterized Preprocessing*. Cambridge University Press, 2019.

[29] R. Ganian, F. Slivovsky, and S. Szeider. Meta-kernelization with structural parameters. *J. Comput. Syst. Sci.*, 82(2):333–346, 2016.

[30] A. C. Giannopoulou, D. Lokshtanov, S. Saurabh, and O. Suchý. Tree deletion set has a polynomial kernel but no $\text{opt}^{o(1)}$ approximation. *SIAM J. Discret. Math.*, 30(3):1371–1384, 2016.

[31] J. Gramm, J. Guo, F. Hüffner, and R. Niedermeier. Graph-modeled data clustering: Exact algorithms for clique generation. *Theory Comput. Syst.*, 38(4):373–392, 2005.

[32] E. C. Hols, S. Kratsch, and A. Pieterse. Elimination distances, blocking sets, and kernels for vertex cover. *SIAM J. Discret. Math.*, 36(3):1955–1990, 2022.

[33] Y. Iwata, K. Oka, and Y. Yoshida. Linear-time FPT algorithms via network flow. In C. Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1749–1761. SIAM, 2014.

[34] B. M. P. Jansen and H. L. Bodlaender. Vertex cover kernelization revisited - upper and lower bounds for a refined parameter. *Theory Comput. Syst.*, 53(2):263–299, 2013.

[35] B. M. P. Jansen and S. Kratsch. A structural approach to kernels for ilps: Treewidth and total unimodularity. In N. Bansal and I. Finocchi, editors, *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, volume 9294 of *Lecture Notes in Computer Science*, pages 779–791. Springer, 2015.

[36] B. M. P. Jansen and B. van der Steenhoven. Kernelization for counting problems on graphs: Preserving the number of minimum solutions. In N. Misra and M. Wahlström, editors, *18th International Symposium on Parameterized and Exact Computation, IPEC 2023, September 6-8, 2023, Amsterdam, The Netherlands*, volume 285 of *LIPIcs*, pages 27:1–27:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[37] B. M. P. Jansen and J. J. H. M. Wulms. Lower bounds for protrusion replacement by counting equivalence classes. *Discret. Appl. Math.*, 278:12–27, 2020.

[38] F. Kammer and A. Sajenko. Space-efficient graph kernelizations. In X. Chen and B. Li, editors, *Theory and Applications of Models of Computation - 18th Annual Conference, TAMC 2024, Hong Kong, China, May 13-15, 2024, Proceedings*, volume 14637 of *Lecture Notes in Computer Science*, pages 260–271. Springer, 2024.

[39] S. Kratsch and P. Kunz. Approximate turing kernelization and lower bounds for domination problems. In N. Misra and M. Wahlström, editors, *18th International Symposium on Parameterized and Exact Computation, IPEC 2023, September 6-8, 2023, Amsterdam, The Netherlands*, volume 285 of *LIPIcs*, pages 32:1–32:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[40] S. Kratsch, D. Marx, and M. Wahlström. Parameterized complexity and kernelizability of max ones and exact ones problems. *ACM Trans. Comput. Theory*, 8(1):1:1–1:28, 2016.

[41] S. Kratsch and M. Wahlström. Preprocessing of min ones problems: A dichotomy. In S. Abramsky, C. Gavoille, C. Kirchner, F. M. auf der Heide, and P. G. Spirakis, editors, *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part I*, volume 6198 of *Lecture Notes in Computer Science*, pages 653–665. Springer, 2010.

[42] S. Kratsch and M. Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. *J. ACM*, 67(3):16:1–16:50, 2020.

[43] D. Lokshtanov, P. Misra, S. Saurabh, and M. Zehavi. Kernelization of counting problems. In V. Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference, ITCS 2024, January 30 to February 2, 2024, Berkeley, CA, USA*, volume 287 of *LIPIcs*, pages 77:1–77:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[44] E. Sam, B. Bergougnoux, P. A. Golovach, and N. Blaser. Kernelization for finding lineal topologies (depth-first spanning trees) with many or few leaves. In H. Fernau and K. Jansen, editors, *Fundamentals of Computation Theory - 24th International Symposium, FCT 2023, Trier, Germany, September 18-21, 2023, Proceedings*, volume 14292 of *Lecture Notes in Computer Science*, pages 392–405. Springer, 2023.

[45] D. M. Thilikos. A retrospective on (meta) kernelization. In F. V. Fomin, S. Kratsch, and E. J. van Leeuwen, editors, *Treewidth, Kernels, and Algorithms - Essays Dedicated to Hans L. Bodlaender on the Occasion of His 60th Birthday*, volume 12160 of *Lecture Notes in Computer Science*, pages 222–246. Springer, 2020.

[46] S. Thomassé. A $4k^2$ kernel for feedback vertex set. *ACM Trans. Algorithms*, 6(2):32:1–32:8, 2010.

[47] C. Xu, L. Dai, and K. Liu. Bloomfilter-based practical kernelization algorithms for minimum satisfiability. In B. Luo, L. Cheng, Z. Wu, H. Li, and C. Li, editors, *Neural Information Processing - 30th International Conference, ICONIP 2023, Changsha, China, November 20-23, 2023, Proceedings, Part IX*, volume 1963 of *Communications in Computer and Information Science*, pages 38–47. Springer, 2023.

A Problem definitions

VERTEX COVER (VC)

- **Type:** Minimization problem.
- **Input:** A graph G.
- **Feasible solutions:** A vertex set $S \subseteq V(G)$ s.t. $G - S$ is an independent set.
- **Solution value:** The size $|S|$ of S.

FEEDBACK VERTEX SET (FVS)

- **Type:** Minimization problem.
- **Input:** A graph G.
- **Feasible solutions:** A vertex set $S \subseteq V(G)$ s.t. $G - S$ is a forest.
- **Solution value:** The size $|S|$ of S.

TREE DELETION SET (TDS)

- **Type:** Minimization problem.
- **Input:** A graph G.
- **Feasible solutions:** A vertex set $S \subseteq V(G)$ s.t. $G - S$ is a tree.
- **Solution value:** The size $|S|$ of S.

DOMINATING SET (DS)

- **Type:** Minimization problem.
- **Input:** A graph G.
- **Feasible solutions:** A vertex set $S \subseteq V(G)$ s.t. $N_G[S] = V(G)$.
- **Solution value:** The size $|S|$ of S.

LONG CYCLE (LC)

- **Type:** Maximization problem.
- **Input:** A graph G.
- **Feasible solutions:** A cycle $C = (v_1, v_2, \ldots, v_r, v_1)$ in G s.t. $v_1, \ldots, v_r$ are distinct vertices from G.
- **Solution value:** The length r of C.

LONG PATH (LP)

- **Type:** Maximization problem.
- **Input:** A graph G.
- **Feasible solutions:** A path $P = (v_1, v_2, \ldots, v_r)$ in G s.t. $v_1, \ldots, v_r$ are distinct vertices from G.
- **Solution value:** The length $r - 1$ of P.

HAMILTONIAN CYCLE (HC)

- **Type:** Decision problem.
- **Input:** A graph G.
- **Question:** Does there exist a cycle C in G that visits all vertices of G?

HAMILTONIAN PATH (HP)

- **Type:** Decision problem.
- **Input:** A graph G.
- **Question:** Does there exist a path P in G that visits all vertices of G?

CLUSTER EDITING (CE)

- **Type:** Minimization problem.
- **Input:** A graph G.
- **Feasible solutions:** An edge set $E \subseteq \{\{u, v\} \mid u, v \in V\}$ s.t. $G \triangle E \coloneqq (V(G), ((E(G) \cup E) \setminus (E(G) \cap E)))$ is a cluster.
- **Solution value:** The size $|E|$ of E.

CLUSTER VERTEX DELETION (CVD)

- **Type:** Minimization problem.
- **Input:** A graph G.
- **Feasible solutions:** A vertex set $S \subseteq V(G)$ s.t. $G - S$ is a cluster.
- **Solution value:** The size $|S|$ of S.

MAXIMUM CUT (MC)

- **Type:** Maximization problem.
- **Input:** A graph G.
- **Feasible solutions:** A partition $X \dot{\cup} Y$ of $V(G)$.
- **Solution value:** The size $|E|$ of $E = \{\{u, v\} \in E(G) \mid u \in X, v \in Y\}$.

NUMBER OF VERTICES WITH DEGREE $\neq 2$ ($\#v, \deg(v) \neq 2$)

- **Type:** Minimization problem.
- **Input:** A graph G.
- **Feasible solutions:** A vertex set $S \subseteq V(G)$ s.t. all vertices in $V(G) \setminus S$ have degree equal two.
- **Solution value:** The size $|S|$ of S.