

# Polynomial Kernel and Incompressibility for Prison-Free Edge Deletion and Completion

Séhane Bel Houari-Durand  

ENS Lyon, France

Eduard Eiben  

Department of Computer Science, Royal Holloway University of London, UK

Magnus Wahlström  

Department of Computer Science, Royal Holloway University of London, UK

---

## Abstract

Given a graph  $G$  and an integer  $k$ , the  $H$ -FREE EDGE DELETION problem asks whether there exists a set of at most  $k$  edges of  $G$  whose deletion makes  $G$  free of induced copies of  $H$ . Significant attention has been given to the *kernelizability* aspects of this problem – i.e., for which graphs  $H$  does the problem admit an “efficient preprocessing” procedure, known as a *polynomial kernelization*, where an instance  $I$  of the problem with parameter  $k$  is reduced to an equivalent instance  $I'$  whose size and parameter value are bounded polynomially in  $k$ ? Although such routines are known for many graphs  $H$  where the class of  $H$ -free graphs has significant restricted structure, it is also clear that for most graphs  $H$  the problem is *incompressible*, i.e., admits no polynomial kernelization parameterized by  $k$  unless the polynomial hierarchy collapses. These results led Marx and Sandeep to the conjecture that  $H$ -FREE EDGE DELETION is incompressible for any graph  $H$  with at least five vertices, unless  $H$  is complete or has at most one edge (JCSS 2022). This conjecture was reduced to the incompressibility of  $H$ -FREE EDGE DELETION for a finite list of graphs  $H$ . We consider one of these graphs, which we dub the *prison*, and show that PRISON-FREE EDGE DELETION has a polynomial kernel, refuting the conjecture. On the other hand, the same problem for the complement of the prison is incompressible.

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

**Keywords and phrases** Graph modification problems, parameterized complexity, polynomial kernelization

**Digital Object Identifier** 10.4230/LIPIcs.STACS.2025.52

**Related Version** *Full Version*: <https://arxiv.org/pdf/2501.15952>

## 1 Introduction

Let  $H$  be a graph. A graph  $G$  is  $H$ -free if it does not contain  $H$  as an induced subgraph. More generally, let  $\mathcal{H}$  be a collection of graphs. A graph is  $\mathcal{H}$ -free if it is  $H$ -free for every  $H \in \mathcal{H}$ . In the  $H$ -FREE EDGE EDITING (respectively  $\mathcal{H}$ -FREE EDGE EDITING) problem, given a graph  $G$  and an integer  $k$ , the task is to add or delete at most  $k$  edges from  $G$  such that the result is  $H$ -free (respectively  $\mathcal{H}$ -free). The EDGE DELETION and EDGE COMPLETION variants are the variants where only deletions, respectively only adding edges is allowed. These are special cases of the much more general *graph modification* problem class, where a problem is defined by a graph class  $\mathcal{G}$  and the natural problem variants ( $\mathcal{G}$  EDGE EDITING/DELETION/COMPLETION respectively  $\mathcal{G}$  VERTEX DELETION) are where the input graph  $G$  is to be modified so that the result is a member of  $\mathcal{G}$ .

As Cai [2] noted, for every finite  $\mathcal{H}$ , the  $\mathcal{H}$ -free graph modification problems are FPT with a running time of  $O^*(2^{O(k)})$  by a simple branching algorithm. However, the question of *kernelization* is much more subtle. A *polynomial kernelization* for a parameterized problem is a polynomial-time procedure that takes as input an instance of the problem, for example,  $I = (G, k)$  in the case of a graph modification problem, and outputs an instance  $(G', k')$  of



© Séhane Bel Houari-Durand, Eduard Eiben, and Magnus Wahlström;  
licensed under Creative Commons License CC-BY 4.0

42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025).

Editors: Olaf Beyersdorff, Michał Pilipczuk, Elaine Pimentel, and Nguyễn Kim Thăng;

Article No. 52; pp. 52:1–52:17



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



the same problem such that  $|V(G')|, k' \leq p(k)$  for some polynomially bounded function  $p(k)$  of  $k$ , and such that  $(G', k')$  is a yes-instance if and only if  $(G, k)$  is a yes-instance. If so, we say that the problem has a *polynomial kernel*. This has been used as a way to capture the notion of efficient instance simplification and preprocessing, and deep and extensive work has been done on determining whether various parameterized problems have polynomial kernels or not (under standard complexity-theoretical assumptions). See the book by Fomin et al. [11].

For many graph modification problems, both those characterized by finite and infinite families  $\mathcal{H}$ , polynomial kernelization is known, but for many others, the question is wide open; see the survey of Crespelle et al. on parameterized edge modification problems [5]. For the structurally simpler case of  $H$ -FREE EDGE DELETION, if  $H$  is a clique then the problem reduces to  $d$ -HITTING SET for  $d = |E(H)|$  and has a polynomial kernel by the sunflower lemma, and if  $|E(H)| \leq 1$  then the problem is trivial. For the same reason,  $H$ -FREE VERTEX DELETION has a polynomial kernel for every fixed  $H$ . But in all other cases, the question is more intricate, since deleting an edge in one copy of  $H$  in  $G$  can cause another copy of  $H$  to occur, implying a dependency between modifications that is not present in the simpler cases. Beyond cliques and near-empty graphs, polynomial kernels are known when  $H$  is  $P_3$  (i.e.,  $H$ -free graphs are cluster graphs) [12],  $P_4$  (i.e.,  $H$ -free graphs are cographs) [13], the paw [7] and the diamond [4] (see Figure 1). Kernels are also known for several simple classes characterized by finite sets  $\mathcal{H}$ . But there are significant open cases; Crespelle et al. [5] highlight the classes of claw-free graphs and line graphs, although the case of line graphs has since been resolved [6]. Initially, progress led Fellows et al. [9] to ask (very speculatively) whether  $\mathcal{H}$ -free graph modification problems have polynomial kernels for all finite  $\mathcal{H}$ . This was refuted by Kratsch and Wahlström [14], and after a series of lower bounds, most importantly by Cai and Cai [3], the answer now appears to be the opposite – the  $\mathcal{H}$ -free graph modification problems have polynomial kernels only for particularly restrictive choices of  $\mathcal{H}$ . Furthermore, in all such cases the kernel depends intimately on the structural characterization of the graph class, such as structural decomposition results. However, it would appear unlikely that such a structural characterization of  $H$ -free graphs should exist for any arbitrary graph  $H$ , and correspondingly, we would expect  $H$ -free edge modification problems not to admit polynomial kernelization. For example, despite the above-mentioned positive results,  $H$ -FREE EDGE DELETION has no kernel for  $H = P_\ell$  where  $\ell \geq 5$ , for  $H = C_\ell$  for  $\ell \geq 4$  or for any  $H$  such that  $H$  or its complement is 3-connected (excepting the trivial cases) [3]. Marx and Sandeep [15] pushed the pendulum in the other direction and conjectured that for graphs  $H$  on at least five vertices, only the above-mentioned immediate kernels exist, conjecturing the following.

► **Conjecture 1** (Conjecture 2 of [15]).  *$H$ -FREE EDGE DELETION does not have a polynomial kernel for any graph  $H$  on at least 5 vertices, unless  $H$  is complete or has at most one edge.*

They showed that this conjecture is equivalent to the statement that  $H$ -FREE EDGE DELETION admits no polynomial kernelization if  $H$  is one of nineteen specific graphs on five or six vertices. They also gave a corresponding conjecture for  $H$ -FREE EDGE EDITING (where  $|E(H)| = 1$  is no longer a trivial case), and the case of  $H$ -FREE EDGE COMPLETION follows by dualization.

In this paper, we refute this conjecture. We study the graph  $H$  shown in Figure 1 (the complement of  $P_3 + 2K_1$ ), which we dub a *prison* (given that it can be drawn as the 5-vertex “house” graph with additional crossbars added). This is the first graph in the set  $\mathcal{H}$  in [15]. We show that PRISON-FREE EDGE DELETION has a polynomial kernel. On the other hand, PRISON-FREE EDGE COMPLETION admits no polynomial kernelization unless the polynomial hierarchy collapses. We leave PRISON-FREE EDGE EDITING open for future work.

## 1.1 Our results

As expected, our result builds on a characterization of prison-free graphs. We then derive the kernelization and lower bound results working over this characterization.

**Prison-free graphs.** The start of our results is a structure theorem for prison-free graphs. To begin with, note that the 4-vertex induced subgraphs of a prison are  $K_4$ , the diamond and the paw; see Figure 1. The structure of diamond-free and paw-free graphs are known: A graph is diamond-free if and only if it is *strictly clique irreducible*, i.e., every edge of the graph lies in a unique maximal clique [17], and a graph is paw-free if and only if every connected component is either triangle-free or complete multipartite [16]. The structure of prison-free graphs generalizes both.

A *complete multipartite graph* with classes  $P_1, \dots, P_m$  is a graph whose vertex set is the disjoint union  $P_1 \cup \dots \cup P_m$  and with an edge  $uv$  for  $u \in P_i$  and  $v \in P_j$  if and only if  $i \neq j$ . Note that a clique is a complete multipartite graph where every part is a singleton. We show the following.

► **Theorem 2.** *A graph  $G = (V, E)$  is prison-free if and only if the following holds: Let  $F \subseteq V$  be an inclusion-wise maximal set such that  $G[F]$  is complete multipartite with at least 4 parts, and let  $v \in V \setminus F$ . Then  $N(v)$  intersects at most one part of  $F$ .*

Furthermore, let  $\text{cmd}_p(G)$  for  $p \in \mathbb{N}$  be the collection of all inclusion-wise maximal  $F \subseteq V(G)$  such that  $G[F]$  is complete multipartite with at least  $p$  classes. We show that  $\text{cmd}_4(G)$  induces a form of partition of the cliques of  $G$ .

► **Corollary 3.** *Let  $G = (V, E)$  be a prison-free graph. The following hold.*

1. *If  $F, F' \in \text{cmd}_4(G)$  are distinct and  $F \cap F' \neq \emptyset$ , then  $F \cap F'$  intersects only one class  $C$  of  $F$  and  $C'$  of  $F'$ , and there are no edges between  $F \setminus C$  and  $F' \setminus C$ . In particular,  $G[F \cap F']$  is edgeless.*
2. *If  $F, F' \in \text{cmd}_4(G)$  with  $F \cap F' = \emptyset$ , then for every  $v \in F'$ ,  $N(v)$  intersects at most one class of  $F$ .*
3. *Let  $e \in E(G)$  be an edge that occurs in at least one  $K_4$  in  $G$ . Then there is a unique  $F \in \text{cmd}_4(G)$  such that  $e$  occurs in  $G[F]$ .*

In particular, every  $K_p$  in  $G$ ,  $p \geq 4$  is contained in a unique  $F \in \text{cmd}_4(G)$ . It also follows that  $\text{cmd}_4(G)$  can be enumerated in polynomial time in prison-free graphs.

**Lower bound for prison-free completion.** We show that PRISON-FREE EDGE COMPLETION is incompressible, i.e., admits no polynomial kernel parameterized by  $k$  unless the polynomial hierarchy collapses. Counterintuitively, this result exploits the property that minimum solutions for the problem can be *extremely expensive* – we can design a sparse graph  $G$  such that every prison-free supergraph of  $G$  contains  $\Theta(n^2)$  edges (for example, by ensuring that the only possible  $\text{cmd}_4$ -decomposition of a prison-free supergraph of  $G$  consists of a single component  $F = V(G)$ ). More strongly, we use this to show an additive *gap hardness* version of the problem.

► **Theorem 4.** *For any  $\varepsilon > 0$ , it is NP-hard to approximate PRISON-FREE EDGE COMPLETION up to an additive gap of  $g = \Theta(n^{2-\varepsilon})$ , even if  $G$  has an edge  $e$  such that  $G - e$  is  $K_4$ -free.*

With this in place, we can proceed with the lower bound using standard methods, using the notion of cross-composition [1, 11]. We follow the method used in previous lower bounds against kernelization of  $H$ -free edge modification problems [14, 3]. Given a list  $I_1, \dots, I_t$  of

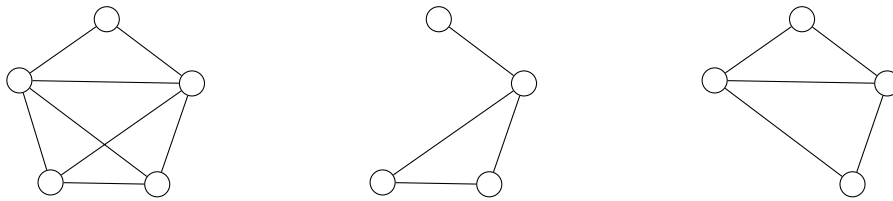
instances of the above gap-version of PRISON-FREE EDGE COMPLETION with parameter value  $k$ , our task is to produce an instance of PRISON-FREE EDGE COMPLETION with parameter  $(k' + \log t)^{O(1)}$  which corresponds to the OR of the instances  $I_i$ . For this, we define a binary tree of height  $O(\log t)$  and place the instances at the leaves of the tree. At the root of the tree, we place a single induced prison. For the internal nodes, we design *propagational gadgets* with the function that if the gadget at the node is edited, then one of the gadgets at the children of the node must be edited as well. Finally, for every instance  $I_j = (G_j, k)$  with an edge  $e_j$  such that  $G_j - e_j$  is prison-free, we connect  $e_j$  to the corresponding gadget at the leaf of the tree and remove  $e_j$  from  $G_j$ . This forces at least one edge  $e_j$ ,  $j \in [t]$  to be added to the resulting graph and the original instance  $I_j = (G_j, k)$  must be solved.

The crux is that, unlike previous proofs (for example in Cai and Cai [3] when  $H$  is 3-connected) we cannot “control” the spread of the edge completion solution to be confined to  $G_j$ . On the contrary, the solution must spread all the way to the root and incorporate all vertices from gadgets on the root-leaf path of the binary tree into a single complete multipartite component of the resulting graph  $G'$ . Thus, we have no tight control over the number of edges added in the corresponding solution. However, by the strong lower bound on gap-hardness of Theorem 4 we do not need tight control – we can simply set the additive gap  $g$  large enough that the number of edges added outside of  $G_j$  in the resulting propagation is a lower-order term compared to  $g$ .

► **Theorem 5.** *PRISON-FREE EDGE COMPLETION does not have a polynomial kernel parameterized by  $k$  unless the polynomial hierarchy collapses.*

**Kernel for prison-free deletion.** The kernelization algorithm depends directly on the structural characterization of prison-free graphs. We start by using the sunflower lemma to obtain a small set  $\mathcal{P}'$  of prisons in the graph  $G$  such that any set of edges of size at most  $k$  that intersects all prisons in  $\mathcal{P}'$  has to intersect all prisons in  $G$ . We let  $S$  be the set of vertices of these prisons. Note that  $|S| = \mathcal{O}(k^8)$ , and outside of  $S$  we only need to be concerned by prisons that are created by deleting some edge from  $G$ . This lets us delete all edges not in  $E(G[S])$  that do not belong to a strict supergraph of a prison. In addition, if a prison in  $G$  contains a single edge inside  $S$ , then such an edge has to be included in every solution of size at most  $k$ , so it can be deleted and  $k$  decreased. After exhaustive application of these two reduction rules, we show that the edges of  $G[V(G) \setminus S]$  can be partitioned into maximal complete multipartite subgraphs of  $G[V(G) \setminus S]$  (even those which do not occur in  $\text{cmd}_4(G[V(G) \setminus S])$ ). For each of these maximal complete multipartite subgraphs, we can check whether it is in some larger complete multipartite subgraph  $F$  in  $G$  that is nicely separated from the rest of  $G$ , in the sense that every  $x \in V(G) \setminus F$  neighbors vertices in at most one class of  $F$ . For any such  $F$ , no supergraph of a prison can contain edge both outside and inside of  $F$  and edges inside of  $F$  can be safely deleted from  $G$ . This allows us to bound the number of maximal complete multipartite subgraphs outside of  $S$  by  $\mathcal{O}(|S|^3)$ . In addition, we show that any supergraph of a prison in  $G$  that contains an edge fully outside of  $S$  is fully contained in  $S \cup F$  for a single maximal multipartite subgraph  $F$  of  $G[V(G) \setminus S]$ . This allows us to treat these multipartite subgraphs separately. Moreover, using the fact that for any edge  $e \in G[S]$ , the graph  $G[V(G) \setminus (S \setminus e)]$  is still prison-free, we can show that the interaction between  $S$  and a maximal multipartite subgraph  $F$  of  $G[V(G) \setminus S]$  is very structured. This allows us to reduce the size of each of these subgraphs and we obtain the following theorem.

► **Theorem 6.** *PRISON-FREE EDGE DELETION admits a polynomial kernel.*



■ **Figure 1** Three graphs: The prison, the paw, and the diamond (as subgraphs of the prison).

**Structure of the paper.** In Section 2 we derive the basic facts about prison-free graphs. Section 3 contains the lower bound against PRISON-FREE EDGE COMPLETION and Section 4 contains the polynomial kernel for PRISON-FREE EDGE DELETION. We conclude in Section 5.

## 2 Structure of prison-free graphs

We begin our study by characterizing the structure of prison-free graphs. This generalizes two structures. The most closely related is for *paw-free graphs*; note that the paw is the subgraph of the prison produced by deleting an apex vertex. Olariu [16] showed that a graph is paw-free if and only if every connected component is either triangle-free or a complete multipartite graph. The paw-free graph modification problems have polynomial kernels due to Eiben et al. [7]. Second, less closely related but still relevant, the *diamond*  $K_4 - e$  is a subgraph of the prison produced by deleting a vertex of degree 3. It is known that a graph is *diamond-free* if and only if every edge occurs in only one maximal clique [17]. The diamond-free graph modification problems have polynomial kernels due to Cao et al. [4]. In a sense, the structure of prison-free graphs generalizes both, as the edge-sets of cliques  $K_p$ ,  $p \geq 4$  in a prison-free graph  $G$  decomposes into complete multipartite induced subgraphs of  $G$ .

We first prove Theorem 2 from the introduction, then use it to derive the more informative Corollary 3.

► **Lemma 7** ( $\star^1$ ). *Let  $G$  be a prison-free graph and let  $K \subseteq V(G)$  induce a  $K_4$  in  $G$ . Then there is a complete multipartite graph  $G[F]$  in  $G$  with  $K \subseteq F$ . Furthermore, if  $F$  is maximal under this condition, then for any  $v \in V(G) \setminus F$ ,  $N(v)$  intersects at most one component of  $F$ .*

Inspired by this, for any graph  $G$  and  $p \geq 2$ , define  $\text{cmd}_p(G)$  to consist of all maximal subsets  $F \subseteq V$  such that  $G[F]$  is complete multipartite with at least  $p$  parts. We refer to  $\text{cmd}_p(G)$  as the *complete multipartite decomposition* of  $G$  (and we will note that it is indeed a decomposition for  $p \leq 4$  if  $G$  is prison-free). The following theorem is now an extension of the previous lemma. Proofs of Theorem 2 and Corollary 3 are deferred to the full version.

► **Theorem 2.** *A graph  $G = (V, E)$  is prison-free if and only if the following holds: Let  $F \subseteq V$  be an inclusion-wise maximal set such that  $G[F]$  is complete multipartite with at least 4 parts, and let  $v \in V \setminus F$ . Then  $N(v)$  intersects at most one part of  $F$ .*

We will use this to derive a more useful description of prison-free graphs.

<sup>1</sup> Results marked with  $\star$  have their proofs deferred to the full version.

► **Corollary 3.** *Let  $G = (V, E)$  be a prison-free graph. The following hold.*

1. *If  $F, F' \in \text{cmd}_4(G)$  are distinct and  $F \cap F' \neq \emptyset$ , then  $F \cap F'$  intersects only one class  $C$  of  $F$  and  $C'$  of  $F'$ , and there are no edges between  $F \setminus C'$  and  $F' \setminus C$ . In particular,  $G[F \cap F']$  is edgeless.*
2. *If  $F, F' \in \text{cmd}_4(G)$  with  $F \cap F' = \emptyset$ , then for every  $v \in F'$ ,  $N(v)$  intersects at most one class of  $F$ .*
3. *Let  $e \in E(G)$  be an edge that occurs in at least one  $K_4$  in  $G$ . Then there is a unique  $F \in \text{cmd}_4(G)$  such that  $e$  occurs in  $G[F]$ .*

In particular, the last item implies that every  $K_p$ ,  $p \geq 4$  is contained in a unique multipartite component  $F \in \text{cmd}_4(G)$ . Since every  $F \in \text{cmd}_4(G)$  contains a  $K_4$ , and each maximal component  $F$  can be found greedily,  $\text{cmd}_4(G)$  can be computed efficiently.

### 3 Incompressibility of Prison-free Edge Completion

In this section, we show that PRISON-FREE EDGE COMPLETION admits no polynomial kernel unless the polynomial hierarchy collapses. The proof is in two parts. First we show a strong inapproximability result – it is NP-hard to approximate PRISON-FREE EDGE COMPLETION within an additive gap of  $g = O(n^{2-\varepsilon})$  for every  $\varepsilon > 0$ , even for graphs with prison-free edge deletion number 1. We then use this to show a cross-composition (see below) for PRISON-FREE EDGE COMPLETION, thereby ruling out polynomial kernels under standard complexity conjectures. This latter part roughly follows the outline of previous proofs of incompressibility [14, 3].

#### 3.1 Initial observations and support gadgets

We begin with some useful statements.

► **Proposition 8.** *For a complete multipartite graph  $K$  with parts of sizes  $a_1, a_2, \dots, a_m$ , the number of edges of  $K$  is  $\frac{1}{2} \sum_{i \neq j} a_i a_j = \frac{1}{2} (|K|^2 - \sum_i a_i^2)$ .*

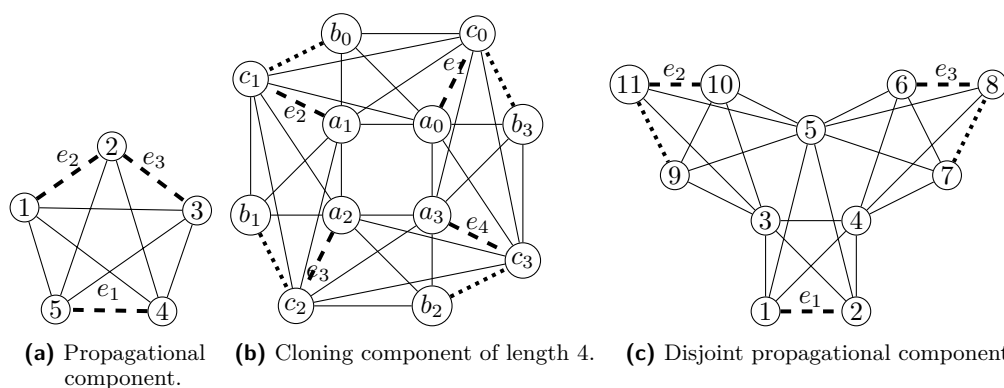
For a graph  $G = (V, E)$  and a set of edges  $A$  over  $V$ , we let  $G \cup A$  denote the graph  $G' = (V, E \cup A)$ . A *prison-free completion set* for  $G$  is an edge set  $A$  over  $V(G)$  such that  $G \cup A$  is prison-free. A *solution* to  $(G, k)$  is a prison-free completion set  $A$  for  $G$  with  $|A| \leq k$ . The following is essential in our lower bounds.

► **Lemma 9** ( $\star$ ). *Let  $G$  be a graph with exactly one induced  $K_4$  and let  $A$  be a minimal prison-free completion set for  $G$ . Then  $\text{cmd}_4(G \cup A)$  has exactly 1 component and  $A$  lives within that component.*

We next show a way to enforce *forbidden edges*, i.e., non-edges  $uv$  in  $G$  such that no prison-free completion set for  $G$  of at most  $k$  edges contains  $uv$ .

► **Lemma 10** ( $\star$ ). *Let  $G$  be a graph,  $k \in \mathbb{N}$ , and  $u, v \in V(G)$  with  $u \neq v$  such that  $uv \notin E(G)$ . There is a graph  $G'$  on vertex set  $V(G') = V(G) \cup F$  such that  $G = G' - F$  and the following holds: the minimal solutions to  $(G', k)$  are precisely the minimal solutions  $A$  to  $(G, k)$  such that  $uv \notin A$ . Furthermore, every solution  $A$  to  $(G', k)$  satisfies  $uv \notin A$ .*

We now proceed with gadget constructions. A *propagational component* is a graph containing 3 distinct non-edges  $e_1, e_2$  and  $e_3$  such that for any graph  $G$  and subset  $A$  of  $V(G)^2$ , if  $G \cup A$  is prison-free and  $e_1 \in A$ , then  $e_2 \in A$  or  $e_3 \in A$ . Figure 2a shows this component. In what follows, we will deduce gadgets from this propagation property, leaving the proof of their prison-freeness for later.



■ **Figure 2** Gadgets for Section 3. Dotted lines are forbidden edges; dashed lines are named “gadget-edges” with special semantics.

► **Definition 11.** Let  $l \geq 4$ . A cloning component of length  $l$  is a component over the vertices  $a_0, \dots, a_{l-1}$ ,  $b_0, \dots, b_{l-1}$ ,  $c_0, \dots, c_{l-1}$  with the edges such that for all  $0 \leq i \leq l-1$ ,  $a_{i+1}, c_{i+1}, b_i, c_i, a_i$  induces a propagational component with  $e_1^i, e_2^i, e_3^i = a_i c_i$ ,  $a_{i+1} c_{i+1}$ ,  $c_{i+1} b_i$ , and the edge  $e_3^i = c_{i+1} b_i$  is forbidden. All arithmetic here is modulo  $l$ .

A cloning component is drawn in Figure 2b. Note that for all  $0 \leq i \leq l-1$ , if a solution  $A$  for an instance  $(G, k)$  contains  $e_1^i$  in a cloning component, then it contains  $e_2^i$  (since  $e_2^i \in A$  or  $e_3^i \in A$ , but  $e_3^i$  is forbidden). We inductively obtain the next property.

► **Lemma 12** (\*). Let  $l \geq 4$ ,  $k \geq 1$  and  $G$  be a graph containing an induced cloning component  $X$  of size  $l$  with vertices named  $a_i$ ,  $b_i$  and  $c_i$  as above. Let  $A$  be a subset of non-edges such that  $|A| \leq k$  and  $G \cup A$  is prison-free. Then either  $\{a_i c_i \mid 0 \leq i \leq l-1\} \cap A = \emptyset$  or  $a_i c_i \in A$  for every  $0 \leq i \leq l-1$ . Furthermore, in the latter case all of  $X$  is contained in a complete multipartite component of  $G \cup A$ .

We define one final gadget, shown in Figure 2c. This does the same job as a propagational component – i.e., if  $e_1 \in A$  then  $e_2 \in A$  or  $e_3 \in A$  – except that the edges  $e_1$ ,  $e_2$ ,  $e_3$  are pairwise vertex-disjoint. This is the *propagational gadget* mentioned in the proof overview.

► **Definition 13.** A disjoint propagational component is a graph isomorphic to the graph shown in Figure 2c, i.e., a graph on a vertex set  $V = \{v_1, \dots, v_{11}\}$  such that vertex sets  $\{v_1, \dots, v_5\}$ ,  $\{v_4, \dots, v_8\}$  and  $\{v_3, v_5, v_9, v_{10}, v_{11}\}$  all induce propagational components,  $v_1 v_2$ ,  $v_3 v_5$ ,  $v_4 v_5$ ,  $v_6 v_8$  and  $v_{10} v_{11}$  are standard non-edges and  $v_7 v_8$  and  $v_9 v_{11}$  are forbidden edges. The edge labelled  $e_1 = v_1 v_2$  is referred to as input edge and the edges  $e_2 = v_{10} v_{11}$  and  $e_3 = v_6 v_8$  are output edges.

### 3.2 NP-hardness of Gap Prison-free Edge Completion

We now prove the first half of the incompressibility result for PRISON-FREE EDGE COMPLETION, namely that it remains NP-hard even in a strong additive gap version.

Let GAP PRISON-FREE EDGE COMPLETION be the variant of PRISON-FREE EDGE COMPLETION where the input is a triple  $(G, k, g)$  and the task is to distinguish between the following cases:

1.  $G$  has a prison-free completion set of at most  $k$  edges.
2.  $G$  has no prison-free completion set of fewer than  $k + g$  edges.

For intermediate cases (where the size  $t$  of a minimum-cardinality prison-free editing set is  $k < t < k + g$ ) the output may be arbitrary. The following is the more precise version of Theorem 4 from the introduction. We defer the construction to the full version of the paper.

► **Theorem 14** (★). *For any  $\varepsilon > 0$ , it is NP-hard to distinguish between yes-instances and no-instances  $(G, k, g)$  of GAP PRISON-FREE EDGE COMPLETION even if  $G$  contains an edge  $e$  such that  $G - e$  is  $K_4$ -free and the gap is  $g = \Theta(n^{2-\varepsilon})$ .*

**Proof sketch.** We can only give a brief sketch here and refer to the full version for details. The result is shown by a reduction from VERTEX COVER on cubic graphs. At its heart is the following principle. We create a graph  $G$  which has a single induced  $K_4$ . Then by Lemma 9, for any minimal prison-free completion set  $A$  of  $G$ ,  $V(A)$  will be contained in a single complete multipartite component  $F$  of  $G \cup A$ . Using forbidden edges, we can have tight control over the partition of  $F$ , which lets us predict  $|A|$  well. In particular, we can ensure that the number of edges  $|A|$  added scales quadratically with  $|V(A)|$ .

Our reduction uses two gadgets: cloning components, of sufficiently large length  $\ell$ , and disjoint propagational components. Let  $(G, t)$  where  $G = (V, E)$  be an input instance of VERTEX COVER where  $G$  is a cubic graph. Using one initial cloning component  $X_0$  with a single seeded active edge  $e$ , we can force the propagation of  $e$  into  $m = |E|$  disjoint activation edges  $e_i$ , one for every edge of  $E$ . For every edge  $ab \in E$ , associated with an edge  $e_i$  of  $X_0$ , we create a disjoint propagational component with input edge  $e_i$  and output edges  $f_{a,i}$  and  $f_{b,i}$  associated with the vertices  $a$  and  $b$  of  $G$ . Finally, for every vertex  $v \in V$  we create a very large cloning component, which contains all edges  $f_{v,j}$  associated with it, and whose length  $\ell$  depends on the desired gap  $g$ . Thus, prison-free completion sets  $A$  of the resulting graph  $G'$  which activate the cloning components of  $s$  distinct vertices of  $G$ , lead to a prison-free supergraph  $G' \cup A$  of  $G'$  where the unique complete multipartite component  $F$  of  $\text{cmd}_4(G' \cup A)$  contains  $O(\ell s)$  vertices, guaranteeing that  $|A| = \Theta((s\ell)^2)$  for a corresponding minimum solution  $A$ . We can now achieve the desired gap by tuning  $\ell$  to be sufficiently large. ◀

### 3.3 Compositionality of Prison-Free Edge Completion

We prove Theorem 5 by an or-composition over instances of GAP PRISON-FREE EDGE COMPLETION, using Theorem 4 to support the composition.

We recall some definitions [1]. A *polynomial equivalence relation* is an equivalence relation on  $\Sigma^*$  such that the following hold:

1. There is an algorithm that given two strings  $x, y \in \Sigma^*$  decides in time polynomial in  $|x| + |y|$  whether  $x$  and  $y$  are equivalent.
2. For any finite set  $S \subset \Sigma^*$ , the number of equivalence classes that  $S$  is partitioned to is polynomially bounded in the size of the largest element of  $S$ .

Let  $L \subseteq \Sigma^*$  be a language,  $\mathcal{R}$  a polynomial equivalence relation and  $Q \subseteq \Sigma^* \times \mathbb{N}$  a parameterized language. An *OR-cross-composition of  $L$  into  $Q$  (with respect to  $\mathcal{R}$ )* is an algorithm that given  $t$  instances  $x_1, \dots, x_t \in \Sigma^*$  of  $L$  belonging to the same equivalence class of  $\mathcal{R}$ , uses time polynomial in  $\sum_{i=1}^t |x_i|$  and outputs an instance  $(y, k)$  of  $Q$  such that the following hold:

1. The parameter value  $k$  is polynomially bounded in  $\max_i |x_i| + \log t$ .
  2.  $(y, k)$  is a yes-instance of  $Q$  if and only if at least one instance  $x_i$  is a yes-instance of  $L$ .
- If an NP-hard language  $L$  has an OR-cross-composition into a parameterized problem  $Q$  then  $Q$  admits no polynomial kernelization, unless the polynomial hierarchy collapses [1].

We proceed to show this for PRISON-FREE EDGE COMPLETION.



We present here a high-level sketch of the result, based directly on Theorem 4. In the full version, we present a more careful proof with explicit parameters. Thus for simplicity, let  $(G_i, k_i, g_i)_{i=1}^t$  be a sequence of instances of GAP PRISON-FREE EDGE COMPLETION with a sufficiently high gap value  $g_i = \Theta(|V(G_i)|^{2-\varepsilon})$ ,  $\varepsilon > 0$ , to be tuned later. Via a polynomial equivalence relation, we may assume that  $|V(G_i)| = n_0$ ,  $|E(G_i)| = m_0$ ,  $k_i = k_0$  and  $g_i = g$  holds for every input instance  $i$ . By Theorem 4, we assume that for every instance  $(G_i, k_i, g_i)$  there is a single edge  $e_i \in E(G_i)$  such that  $G_i - e_i$  is prison-free; refer to this as the *activation edge* of  $G_i$  and delete it from the graph.

For the composition, let  $h \in \mathbb{N}$  be such that  $2^{h-1} < t \leq 2^h$ . Define a balanced binary tree of height  $h$  whose leaves are labelled  $L_i$ ,  $i = 1, \dots, t$ . Place a disjoint propagational component for every internal node  $x$  of the tree, identifying the two output edges with the children of  $x$  and the input edge at  $x$  with the corresponding output edge of the parent of  $x$ . Initially, all input and output edges are absent except that the input edge at the root of the tree is present. Finally identify the output edge leading into the leaf  $L_i$  with the activation edge  $e_i$  of the graph  $G_i$ . If there are any unused output edges, for example if  $t$  is odd, make these edges forbidden. Let  $(G, k)$  be the resulting instance of PRISON-FREE EDGE COMPLETION where  $k$  is yet to be determined.

► **Lemma 15** ( $\star$ ). *Let  $A$  be a prison-free completion set for  $G$ . Then there is some  $i \in [t]$  such that  $A$  contains the activation edge  $e_i$  of the graph  $G_i$ . Furthermore, for every  $i \in [t]$ , any minimal prison-free edge completion set  $A_i$  for  $G_i + e_i$  can be completed into a prison-free completion set  $A$  for  $G$  which contains every output edge in the path from the root to  $L_i$  but no other output edges from the tree.*

We are now ready to finish the proof.

► **Theorem 5.** *PRISON-FREE EDGE COMPLETION does not have a polynomial kernel parameterized by  $k$  unless the polynomial hierarchy collapses.*

**Proof sketch.** We show that the construction above is a cross-composition into PRISON-FREE EDGE COMPLETION with parameter  $k$ . Consider briefly the two cases. First, if for some  $i \in [t]$  the input instance  $(G_i, k_0, g_0)$  is positive, let  $A_i$  be a prison-free completion set for  $G_i$  with  $|A_i| \leq k$ . By Lemma 15 there is a prison-free completion set  $A \supseteq A_i$  for  $G$  that modifies edges along the root-leaf path to  $L_i$  of the binary tree, and does not contain any other output edge of the tree. By Lemma 9 and minimality of  $A$  and  $A_i$  we may now assume that  $A$  touches no vertices of the binary tree except on the root-leaf path to  $L_i$ , and contains no further edges inside  $G_i$  beyond  $A_i$ . Let  $n_t$  be the number of vertices in the gadgets of the tree incident with edges of  $A$ ; since the tree has height  $h = O(\log t)$  and each node is constant size, we have  $n_t = O(\log t)$ . Thus

$$|A| \leq |A_i| + (n_h + n_0)n_h \leq k_0 + O((n_0 + \log t) \log t).$$

On the other hand, assume that for every  $i \in [t]$  the minimum prison-free completion set  $A_i$  for  $G_i$  has  $|A_i| \geq k_0 + g_0$ . Let  $A$  be a prison-free completion set for  $G$ . By Lemma 15 there is some  $i \in [t]$  such that the activation edge  $e_i$  of  $G_i$  is contained in  $A$ . Thus  $A$  contains a prison-free completion set for  $G_i$  and  $|A| \geq k_0 + g_0$ . By the construction of Theorem 4, we can tune the parameters so that these quantities separate, and choose a parameter  $k$  where

$$k_0 + O((n_0 + \log t) \log t) \leq k < k_0 + g_0$$

in which case the reduction is complete. ◀

## 4 Polynomial kernel for Prison-free Edge Deletion

In this section we will find a polynomial kernel for PRISON-FREE EDGE DELETION. Let  $G$  be a graph and  $k \geq 1$ . We will fix  $(G, k)$  throughout this section to be an instance of PRISON-FREE EDGE DELETION.

Throughout this section, for a graph  $G$ , and an edge  $e = uv$  of  $G$ , we call common neighborhood of  $e$  the set  $N_G(e) = N_G(u) \cap N_G(v)$ . In addition, given a graph  $G$  and a set of vertices  $S \subseteq V(G)$ , we denote by  $\bar{S}$  the set  $V(G) \setminus S$ , when  $G$  is clear from the context.

### 4.1 Finding a Small Vertex Modulator

We start by finding a small subset of vertices  $S$  such that any edgeset  $A \subseteq E(G)$  with at most  $k$  edges that intersects all prisons in  $G[S]$  also intersects all prisons in  $G$ . While this is not sufficient for a kernel, as deleting an edge can create a prison, outside of this set, we only need to focus on prisons that are created by deleting an edge. To obtain this set, we will use well known Sunflower Lemma due to Erdős and Rado [8].

A *sunflower* in a set family  $\mathcal{F}$  is a subset  $\mathcal{F}' \subseteq \mathcal{F}$  such that all pairs of elements in  $\mathcal{F}'$  have the same intersection called *core*.

► **Lemma 16** (Sunflower Lemma, [8, 10]). *Let  $\mathcal{F}$  be a family of subsets of a universe  $U$ , each of cardinality exactly  $b$ , and let  $a \in \mathbb{N}$ . If  $|\mathcal{F}| \geq b!(a-1)^b$ , then  $\mathcal{F}$  contains a sunflower  $\mathcal{F}'$  of cardinality at least  $a$ . Moreover,  $\mathcal{F}'$  can be computed in time polynomial in  $|\mathcal{F}|$ .*

► **Lemma 17** ( $\star$ ). *We can in polynomial time either determine that  $(G, k)$  is no-instance of PRISON-FREE EDGE DELETION or compute a set  $S \subseteq V(G)$  with  $|S| \leq 5 \cdot 8! \cdot (k+1)^8$  such that for every prison  $P$  in  $G$  and every  $A \subseteq E(G)$  with  $|A| \leq k$ , it holds that if  $G[S] \Delta A$  is prison-free, then  $A \cap E(G[P]) \neq \emptyset$ .*

**Proof Sketch.** The proof is a straightforward application of Lemma 16. Let  $\mathcal{S} = \{E(P) \mid P \text{ is a prison in } G\}$ . Note that each set  $X \in \mathcal{S}_0$  contains precisely 8 edges. We iteratively apply Lemma 16 on  $\mathcal{S}$  to find a sunflower of size  $k+2$ . If the core of the sunflower  $\mathcal{S}'$  is empty, then  $G$  contains  $k+2$  edge-disjoint prisons and  $(G, k)$  is no instance. Else any solution of size at most  $k$  intersects the core and that is the case even if we require to hit only  $k+1$  prisons of  $\mathcal{S}'$ , so we remove arbitrary prison from  $\mathcal{S}$  and repeat the procedure until no sunflower of size  $k+2$  can be found. At that point  $\mathcal{S}$  contains at most  $8! \cdot (k+1)^8$  many prisons and any  $A \subseteq E(G)$  with  $|A| \leq k$  that intersect all of the prisons that are left in  $\mathcal{S}$  intersects all prisons in  $G$ . We let  $S$  to be the set of all vertices in these prisons. ◀

For the rest of the section and of the proof, we let  $S$  be the set computed by Lemma 17. It follows, as long as we keep  $G[S]$  as the subgraph of the reduced instance, we only need to be concerned about the prisons that are created by removing some edge from  $G$ , as all the prisons that were in  $G$  to start with are hit by a set  $A$  of at most  $k$  edges as long as  $G[S] \Delta A$  is prison-free. Given the above, the following two reduction rules are straightforward.

► **Reduction Rule 1** ( $\star$ ). *If an edge  $e \in (E(G) \setminus E(G[S]))$  is not in a strict supergraph of a prison, delete it.*

► **Reduction Rule 2** ( $\star$ ). *For every prison  $P$  in  $G$ , if  $E(G[S]) \cap P = \{e\}$ , then remove  $e$  and decrease  $k$  by one.*

Thanks to these Reduction Rules, we found a set  $S$  of vertices of size polynomial in  $k$  such that for any subset  $S'$  of vertices of  $S$  such that  $G[S']$  has at most one edge,  $G[\bar{S} \cup S']$  is prison-free. We note that we assume that all reduction rules are applied exhaustively;

that is whenever a reduction rule is applicable, we apply it and restart the process from the beginning. Hence, in all statements in the rest of the section, we implicitly assume that none of the reduction rules can be applied.

## 4.2 Consequences on the Structure of $G[\bar{S}]$

Now we are able to show the following properties that will be useful for the kernel. The following lemma gives us a stronger property than just the characterisation of prison-free graphs.

► **Lemma 18** ( $\star$ ). *Let  $F \in \text{cmd}_3(G[\bar{S}])$  be a maximal complete multipartite subgraph of  $G[\bar{S}]$  such that  $F$  has exactly three classes. Then there exists  $s \in S$  with  $F \subseteq N(s)$ . Moreover, there is a strict supergraph of a prison that contains  $s$  and an edge in  $F$ .*

**Proof Sketch.** Since  $F$  has three classes, there are  $u, v, w$  such that  $uvw$  is a triangle in  $G[\bar{S}]$ . Due to Reduction Rule 1, each edge of  $F$  is in a strict supergraph of a prison and hence in  $K_4$  in  $G$ . Now given a  $K_4$   $(a, b, u, v)$  that contains  $uv$ , we conclude that  $w$  is adjacent to either  $a$  or  $b$ , else  $(a, b, u, v, w)$  induces a prison with at most one edge in  $S$  and Reduction Rule 2 applies. Hence  $\{u, v, w\}$  is a subset of a  $K_4$ , say  $(u, v, w, s)$ . As a consequence of Lemma 7 and since  $G[\bar{S}]$  is prison-free, one can show that  $F$  has to be fully included in the  $K_4$ -free part of  $G[\bar{S}]$ , and hence  $s \in S$ . Now, any vertex  $x \in F \setminus \{u, v, w\}$  is adjacent to exactly two vertices in  $\{u, v, w\}$  and hence if  $sx \notin E(G)$ , then  $(u, v, w, s, x)$  induces a prison without an edge in  $S$ , which is impossible due to the construction of  $S$ . ◀

► **Lemma 19** ( $\star$ ). *For all  $F \in \text{cmd}_3(G[\bar{S}])$ , for all  $x \in \bar{S} \setminus F$ ,  $N_G(x)$  intersects at most one class of  $F$ .*

**Proof Sketch.** The Lemma follows for  $F \in \text{cmd}_4(G[\bar{S}])$  by Lemma 7. Hence, we can assume  $F \in \text{cmd}_3(G[\bar{S}]) \setminus \text{cmd}_4(G[\bar{S}])$ . For the sake of contradiction, assume that for  $x \in \bar{S} \setminus F$ , we have that  $N(x) \cap F$  contains an edge  $uv$ . Since  $F$  has three classes, for every  $w$  in the class  $C$  that does not contain  $u$  nor  $v$ , we have that  $uvw$  is a triangle. Moreover, similarly as in the proof of Lemma 18, one can show using Lemma 7 that  $(u, v, w, x)$  cannot be  $K_4$  and so  $wx \notin E(G)$ . Since,  $F$  is inclusion maximal, there is  $a \in F \setminus C$  such that  $xa \notin E(G)$ . By Lemma 18, there is  $s \in S$  with  $F \subseteq N(s)$ . But then either  $(u, v, w, s, x)$  or  $(u, v, a, s, x)$  induces a prison with no edge in  $G[S]$  (depending on whether  $sx \in E(G)$  or not), which is impossible due to the construction of  $S$ . ◀

► **Lemma 20** ( $\star$ ). *If a supergraph  $P$  of a prison of  $G$  has an edge in  $F \in \text{cmd}_3(G[\bar{S}])$ , then  $P \subseteq S \cup F$ .*

**Proof Sketch.** For the sake of contradiction, let  $P$  be a supergraph of a prison of  $G$  that has an edge  $uv$  in  $F \in \text{cmd}_3(G[\bar{S}])$  and there is  $x \in P \setminus (S \cup F)$ . One can show that due to Lemma 19 and application of Reduction Rule 2,  $P = (a, b, u, v, x)$  is a strict supergraph of a prison with  $\{a, b\} \subseteq S$  and  $vx$  (or  $ux$ ) being the only non-edge of  $P$ . Moreover,  $F$  contains triangle  $uvw$ , as it has at least three classes and  $wx \notin E(G)$  by Lemma 19. Due to Reduction Rule 2,  $(u, v, w, a, b)$  cannot induce a prison, and either  $wa \in E(G)$  or  $wb \in E(G)$ . If  $wa \in E(G)$  (resp.  $wb \in E(G)$ ), then  $(x, a, u, w, v)$  (resp.  $(x, b, u, w, v)$ ) induces a prison in  $G$  with at most one edge in  $G[S]$ , contradicting application of Reduction Rule 2. ◀

Let's now focus on the edges of  $\bar{S}$  that are not in any triangle. We show that since Reduction Rules 1 and 2 has been exhaustively applied, even these edges can be partitioned to maximal complete bipartite subgraphs.

## 52:12 Kernelization for Prison-Free Edge Deletion and Completion

Let  $B$  be the set of edges of  $G[\bar{S}]$  that are not in any triangle in  $G[\bar{S}]$ . For all  $e = ab \in E(G[S])$ , we note  $B_e = B \cap \{uv : u, v \in N(a) \cap N(b)\}$ . Note that every edge  $f \in B$  belongs to a  $K_4$  in  $G$  due to the application of Reduction Rule 1. Since,  $f$  is not in any triangle in  $G[\bar{S}]$ , it follows that it is in some  $B_e$  for  $e \in E(G[S])$ . On the other hand, we show that if  $B_{e_1} \cap B_{e_2} \neq \emptyset$ , then  $B_{e_1} = B_{e_2}$ , otherwise  $G$  contains a prison with at most one edge in  $G[S]$ , which gives us the following lemma.

► **Lemma 21** ( $\star$ ).  $\{B_e \mid e \in E(G[S])\}$  is a partition of  $B$ .

The following lemma is a straightforward consequence of Reduction Rule 2, since for every  $e \in E(S)$ ,  $G[\bar{S} \cup e]$  is prison-free and hence  $N_G(e) \cap \bar{S}$  is  $\bar{P}_3$ -free.

► **Lemma 22** ( $\star$ ). For all  $e \in E(G[S])$ ,  $N_G(e) \cap \bar{S}$  is complete multipartite and if  $B_e$  is non empty,  $N_G(e)$  is a complete bipartite graph.

► **Lemma 23** ( $\star$ ). Let  $e \in E(G[S])$ . Assume that  $B_e$  is not empty. Then any induced supergraphs of a prison  $P$  that has an edge in  $B_e$  is in  $B_e \cup S$ .

**Proof Sketch.** Assume that there is an induced supergraphs of a prison  $P = (a, b, u, v, w)$ , where  $uv$  in  $B_e$  and  $w \in \bar{S} \setminus B_e$ . It follows from Reduction Rule 2 and the fact that  $uv$  is not in a triangle in  $G[\bar{S}]$  that (1)  $P$  is a strict supergraph of a prison with only non-edge  $uw$  (resp.  $vw$ ) and (2)  $\{a, b\} \subseteq S$ . By Lemma 21,  $B_e = B_{ab}$ . Since  $B_e$  is maximal complete bipartite subgraph of  $G[\bar{S}]$ , the class that contains  $v$  (resp.  $u$ ), contains a vertex  $v'$  (resp.  $u'$ ) with  $v'w \notin E(G)$ . But then  $(a, b, v, v', w)$  (resp.  $(a, b, u, u', w)$ ) induces a prison in  $G$  with only one edge inside  $S$ , which is a contradiction. ◀

It follows that we can partition the edges of  $G[\bar{S}]$  in complete multipartite subgraphs, where one of these complete multipartite subgraphs can intersect another in at most one class. The following reduction rule lets us reduce the number of complete multipartite subgraphs in this partition to  $|S|^3 + |S|^2 = \mathcal{O}(k^{24})$ . Given this bound, we will reduce size of each of these components as well as number of isolated vertices in  $G[\bar{S}]$  by a polynomial function in  $k$  as well.

► **Reduction Rule 3** ( $\star$ ). Let  $F' \in \text{cmd}_3(G[\bar{S}])$ . If  $F \subseteq V(G)$  is a maximal complete multipartite component such that  $F \supseteq F'$  and for every vertex  $v \notin F$ ,  $N(v)$  intersects at most one class of  $F$ , remove all edges of  $F$ .

**Proof Sketch.** It is not difficult to show that there is no supergraph of a prison in  $G$  that contains an edge with both endpoints in  $F$  and at the same time an edge with at least one endpoint outside of  $F$ . Given this for every  $A \subseteq E(G)$ , every prison in  $G \Delta A$  is either all edge in  $F$  or all edges in  $G - E(F)$ , since  $F$  is complete multipartite and hence prison-free, it suffices to hit all prisons in  $G - E(F)$ . ◀

Recall that we always assume that none of the previous reduction rules can be applied, in particular from now on we assume also that Reduction Rule 3 is not applicable.

From now on, we denote  $\mathcal{F} = \text{cmd}_3(G[\bar{S}]) \cup \{B_e \mid e \in E(G[S])\}$ .

► **Lemma 24** ( $\star$ ). The edges of  $G[\bar{S}]$  can be partitioned into at most  $|S|^3 + |S|^2$  many maximal complete multipartite subgraphs of  $G[\bar{S}]$ .

**Proof Sketch.** Clearly every edge in  $G[\bar{S}]$  is in a maximal complete multipartite subgraph of  $G[\bar{S}]$ . We only need to show that each edge is in precisely one such subgraph and that their number is at most  $|S|^3 + |S|^2$ . First note the edges in  $B$  are partitioned into at most

$|E(G[S])| \leq |S|^2$  many complete bipartite graphs by Lemma 21. Hence, we only need to consider the edges that are in some  $F \in \text{cmd}_3(G[\bar{S}])$ . It is a rather straightforward consequence of Lemma 19 that any edge  $e$  in  $F$  cannot be in any other maximal complete multipartite subgraph in  $\text{cmd}_3(G[\bar{S}])$ . It remains to show that  $|\text{cmd}_3(G[\bar{S}])| \leq |S|^3$ .

We will now define a function  $f : \text{cmd}_3(G[\bar{S}]) \rightarrow \text{cmd}_4(G)$  such that for all  $F \in \text{cmd}_3(G[\bar{S}])$ , it holds that (1)  $F \subseteq f(F)$  and (2)  $f(F)$  contains at least one vertex in  $S$ . That is  $f$  is injective. We will show that every edge in  $G[S]$  is in at most  $|S|$  many complete multipartite subgraph in the image of  $f$  and every element of the image of  $f$  contains an edge of  $G[S]$ , bounding  $\text{cmd}_3(G[\bar{S}])$  by  $|E(G[S])| \leq |S|^3$ .

If  $F \in \text{cmd}_4(G[\bar{S}])$ , then since Reduction Rule 3 has been exhaustively applied, there is  $v \in S$  such that  $N(v)$  intersects at least two classes of  $G[F]$ , and since  $G[\bar{S} \cup \{v\}]$  is prison-free and  $F$  contains a  $K_4$ ,  $G[F \cup \{v\}]$  is complete multipartite. We define  $f(F)$  as any maximal complete multipartite component of  $G$  containing  $F \cup \{v\}$ . Else, if  $F \in \text{cmd}_3(G[\bar{S}]) - \text{cmd}_4(G[\bar{S}])$ , then by Lemma 18, there is a vertex  $s \in S$  such  $F \subseteq N(s)$  and  $G[F \cup \{s\}]$  is a complete multipartite graphs with at least four parts. We define  $f(F)$  as a maximal complete multipartite subgraph of  $G$  containing  $F \cup \{s\}$ .

▷ **Claim 25 (★).** Let  $e = uv$  be an edge of  $G[S]$ . Then at most  $|S|$  elements of  $\text{Im}(f)$  contains the vertices of  $e$ .

There are thus at most  $|S|^3$  elements of  $\text{Im}(f)$  that contains an edge in  $S$ .

▷ **Claim 26 (★).** Each element of  $\text{Im}(f)$  contains an edge in  $S$ .

So  $|\text{Im}(f)| \leq |S|^3$ . Since  $f$  is injective,  $|\text{cmd}_3(G[\bar{S}])| \leq |S|^3$ . Therefore,  $|\mathcal{F}| = |\text{cmd}_3(G[\bar{S}])| + |\{B_e \mid e \in E(G[S])\}| \leq |S|^3 + |S|^2$ . ◀

We will have a kernel once we have bounded the size of a maximal complete multipartite subgraphs and the number of isolated vertices in  $G[\bar{S}]$ . Before we show how to bound those, let us observe the following two simple lemmas.

▶ **Lemma 27 (★).** Let  $F \in \text{cmd}_3(G[\bar{S}])$  and  $s \in S$ . If  $N(s)$  intersects more than one class of  $F$ , then either  $N(s) \cap F = F \setminus C$ , where  $C$  is a single class in  $F$ , or  $N(s) \cap F = F$ .

▶ **Lemma 28 (★).** Let  $e$  such that  $B_e$  is non empty, and let  $s \in S$ .  $N(s)$  either contains  $B_e$  or it intersects at most one class of  $G[B_e]$ .

### 4.3 Marking important vertices

We will now mark some important vertices that we will keep to preserve the solutions and afterwards argue that we can remove all the remaining vertices without changing the value of an optimal solution. To better understand why this marking procedure works, it is useful to recall Lemmas 20 and 23 that state that any (not necessarily strict) supergraph of a prison that has at least one edge in some  $F \in \mathcal{F}$  is fully contained in  $S \cup F$ . That is only supergraphs of the prison that can contain vertices in more than one complete multipartite subgraph from  $\mathcal{F}$  are those with all edges either in  $S$  or between  $S$  and  $N(S)$ . Such supergraphs of a prison have always at most two vertices outside of  $S$ . Before we give the details of the marking procedure, let us introduce a concept of *neighborhood pattern*. Let  $X \subseteq V(G)$  and  $a \in V(G) \setminus X$ , then the neighborhood pattern of  $a$  in  $X$  is  $N(a) \cap X$ . Moreover, for two vertices  $a, b \in V(G) \setminus X$ , we say  $a$  and  $b$  have the same *neighborhood pattern* in  $X$  if  $N(a) \cap X = N(b) \cap X$ .

## 52:14 Kernelization for Prison-Free Edge Deletion and Completion

Now, let  $F \in \mathcal{F}$ . By Lemmas 27 and 28, for every  $s \in S$ , if  $N(s) \cap F \neq \emptyset$ , then there are only three possibilities how  $N(s)$  and  $F$  can interact. Namely,

1.  $N(s) \cap F \subseteq C$  for a single part  $C$  of  $F$ ;
2.  $N(s) \cap F = F \setminus C$  for a single part  $C$  of  $F$ ;
3.  $N(s) \cap F = F$ .

Given the above, we can split the classes  $C$  of  $F$  into two types.

**Type 1.** There is  $s \in S$  such that  $N(s) \cap F \subseteq C$  or  $N(s) \cap F = F \setminus C$ . We denote the set of classes of Type 1. as  $\mathcal{T}_F^1$ ;

**Type 2.** The rest, which we denote  $\mathcal{T}_F^2$ .

Note that  $|\mathcal{T}_F^1| \leq |S|$  due to Lemmas 27 and 28 all classes  $C$  in  $F$  with  $C \notin \mathcal{T}_F^1$  have exactly the same neighborhood in  $S$ .

We compute a set  $M_F$  of marked vertices for the component  $F \in \mathcal{F}$  as follows. We note that whenever we say, we mark some number  $x$  of vertices with some property, we mean that if there are more than  $x$  many vertices with that property, we mark arbitrary  $x$  many of them, else we mark all of them. Let us start with marking vertices in a class  $C$  of Type 1. for every  $C \in \mathcal{T}_F^1$ . For each  $S' \subseteq S$  with  $|S'| = 4$ , and for each neighborhood pattern  $\xi$  in  $S'$ , we add to  $M_F$  arbitrary  $2k + 5$  vertices of  $C$  with the neighborhood pattern  $\xi$  in  $S'$ . Observe that for any  $S''$  with  $|S''| < 4$ , there is  $S' \supset S''$  with  $|S'| = 4$  and so for any neighborhood pattern  $\xi''$  in  $S''$ , there is a neighborhood pattern in  $S'$  that is equal to  $\xi''$  if restricted to  $S''$ . Hence, using this marking, we marked at least  $2k + 5$  vertices with any neighborhood pattern in any  $S'$  with  $|S'| \leq 4$  as well.

In addition, we pick arbitrary  $2k + 5$  classes of  $F$  that are in  $\mathcal{T}_F^2$  and add arbitrary  $2k + 5$  vertices from each picked class to  $M_F$ .

► **Lemma 29** ( $\star$ ). *Given the above marking procedure, for all  $F \in \mathcal{F}$ , it holds that  $|M_F| \leq |S|^5 \cdot (2k + 5) + (2k + 5)^2$ .*

In addition to marking the set  $M_F$  for each  $F \in \mathcal{F}$ , we mark additional set of at most  $2^4 \cdot \binom{|S|}{4} \cdot (2k + 5)$  vertices by going over all subsets  $S'$  of  $S$  of size 4 and for each neighborhood pattern  $\xi$  in  $S'$ , we mark at most  $2k + 5$  additional vertices of  $V(G) \setminus S$  that are not in any  $F \in \mathcal{F}$  with the given neighborhood pattern  $\xi$  in  $S'$ . We denote this set of at most  $|S|^4 \cdot (2k + 5)$  vertices as  $M_\emptyset$ . Additionally, observe that this way, we mark at least  $2k + 5$  vertices for each neighborhood pattern in any subset of  $S$  of size at most four as well.

► **Lemma 30** ( $\star$ ). *Let  $G' = G[S \cup M_\emptyset \cup \bigcup_{F \in \mathcal{F}} M_F]$ . Then  $(G, k)$  is yes-instance of PRISON-FREE EDGE DELETION if and only if  $(G', k)$  is. In addition  $|V(G')| = \mathcal{O}(k^{65})$ .*

**Proof.** The bound on the size of  $G'$  follows from the fact that that  $|S| = \mathcal{O}(k^8)$ ,  $|\mathcal{F}| \leq |S|^3 + |S|^2$ , and Lemma 29. Now,  $G'$  is an induced subgraph of  $G$  and hence for every  $A \subseteq E(G)$ , if  $G \Delta A$  is prison-free, then also  $G' \Delta (A \cap E(G'))$  is. Therefore, if  $(G, k)$  is yes-instance, then so is  $(G', k)$ .

For the rest of the proof, let us assume that  $(G', k)$  is yes-instance and let  $A \subseteq E(G')$  be such that  $|A| \leq k$  and  $G' \Delta A$  is prison-free. We show that  $G \Delta A$  is also prison-free. For the sake of contradiction, let's assume that  $P = (a, b, c, d, e)$  induces a prison in  $G \Delta A$ . Let us distinguish two cases depending on whether  $P$  contains an edge between two vertices outside of  $S$  or not.

**Case 1.** All edges of  $P$  have at least one endpoint in  $S$ . Note that in this case, there are at most two vertices of  $P$  outside of  $S$ .

If only one vertex of  $P$ , say  $a$ , is outside of  $S$ , then clearly  $a$  is the only vertex of  $P$  not in  $G'$ . Note that this also means that none of the edges incident with  $a$  is in  $A$ . Moreover,

$G'$  contains at least  $2k + 5$  vertices with the same neighborhood pattern in  $\{b, c, d, e\}$ , else  $a$  would have been either in  $M_\emptyset$  or in  $M_F$  for some  $F \in \mathcal{F}$ . Since  $|A| \leq k$ , it follows that at least one of these  $2k + 5$  vertices, let's call it  $a'$ , is not incident to an edge in  $A$  and in particular to an edge between  $a'$  and a vertex in  $\{b, c, d, e\}$ . However, then  $(G' \Delta A)[\{a', b, c, d, e\}]$  is isomorphic to  $P$  and induces a prison, which is a contradiction with  $G' \Delta A$  being prison-free.

Now assume that two vertices  $a$  and  $b$  are outside of  $S$ , and there is no edge between  $a$  and  $b$  in  $G \Delta A$ . Note that the other non-edge of  $P$  share an endpoint with  $ab$ , and w.l.o.g., we can assume it is  $ac$ . As at least one of  $a$  and  $b$  is not in  $G'$ ,  $ab \notin E(G)$ . Moreover,  $ac \in E(G) \cap A$ , else  $P$  is a prison in  $G$  and Lemma 17 implies that  $A$  intersects  $P$ . Hence,  $b \notin V(G')$ , by our marking procedure, there are at least  $2k + 5$  vertices  $b'$  in  $V(G') \setminus S$  with  $\{c, d, e\} \subseteq N(b')$  and  $bb' \notin E(G)$ . One of these vertices is not incident on any edge in  $A$ . For such a vertex  $b'$ , either  $(a, b', c, d, e)$  or  $(a, b, b', d, e)$  is a prison in  $G' \Delta A$  depending on whether  $ab' \in E(G)$  or  $ab' \notin E(G)$ .

**Case 2.**  $P$  contains an edge in  $G - S$ . Then by Lemmas 20 and 23, it follows that there exists  $F \in \mathcal{F}$  such that  $P \subseteq S \cup F$ . Let  $S' = P \cap S$ . Now, for  $x \in P \setminus S$ , if  $x$  belongs to  $C \in \mathcal{T}_F^1$ , then by construction of  $M_F$  and the fact that  $|A| \leq k$ ,  $M_F$  either contains  $x$  or a vertex  $x'$  such that (1)  $x' \in C$ , (2)  $N(x) \cap S' = N(x') \cap S'$ , and (3)  $x'$  is not incident to any edge of  $A$ . On the other hand, if some of the vertices in  $P \cap F$  are in the classes in  $\mathcal{T}_F^2$ , then all such vertices have exactly same neighborhood in  $S$ , moreover, either all their respective classes contain vertices in  $M_F$ , or there are at least five classes of  $F$  that each has  $2k + 5$  vertices in  $G'$  and no vertex in these classes is incident on an edge in  $A$ . Hence, for each vertex  $x$  in  $P \cap F$  that belongs to a class  $C \in \mathcal{T}_F^2$ , we can find  $C' \in \mathcal{T}_F^2$  and  $x' \in C'$  such that (1)  $x' \in M_F$ , (2)  $x'$  is not incident on any edge in  $A$ , and (3) if  $x$  and  $y$  are in  $P \cap F$ , then  $x'$  and  $y'$  are in the same class of  $F$  if and only if  $x$  and  $y$  are. Let  $P' = (a', b', c', d', e')$  be a subgraph of  $G'$  such that for all  $x \in \{a, b, c, d, e\}$ , if  $x \in G'$ , then  $x' = x$  and else  $x'$  is computed as described above, depending on whether  $x \in \mathcal{T}_F^1$  or  $x \in \mathcal{T}_F^2$ . It follows that for all  $x, y \in \{a, b, c, d, e\}$ , there is an edge  $xy \in E(G)$  if and only if  $x'y' \in E(G')$ . Moreover, since  $xy \in A$  implies that  $\{x, y\} \subseteq V(G')$ , it follows that  $xy \in A$  if and only if  $x'y' \in A$ . Therefore,  $P'$  induces a prison in  $G'$ , which is a contradiction.

Hence, such prison  $P$  in  $G \Delta A$  cannot exist and  $G \Delta A$  is prison-free as well. Consequently,  $(G', k)$  is yes-instance of PRISON-FREE EDGE DELETION if and only if  $(G, k)$  is and the Lemma follows.  $\blacktriangleleft$

The polynomial kernel for PRISON-FREE EDGE DELETION then follows from Lemma 30 by observing that all reduction rules as well as marking procedure can be applied in polynomial time.

► **Theorem 6.** *PRISON-FREE EDGE DELETION admits a polynomial kernel.*

## 5 Conclusions

We have showed that  $H$ -FREE EDGE DELETION has a polynomial kernel when  $H$  is the 5-vertex graph we call the “prison” (consisting of  $K_5$  minus two adjacent edges). On the other hand,  $H$ -FREE EDGE COMPLETION for the same graph  $H$  does not have a polynomial kernel unless the polynomial hierarchy collapses. By edge complementation, this is equivalent to the statement that  $\overline{H}$ -FREE EDGE DELETION has no polynomial kernel, where  $\overline{H}$  is the edge complement of  $H$ . The positive result refutes a conjecture by Marx and Sandeep [15], who conjectured that  $H$ -FREE EDGE DELETION has no polynomial kernel for any graph  $H$  on at least five vertices except trivial cases.

In [15], the conjecture is reduced to the statement that  $H$ -FREE EDGE DELETION has no polynomial kernel for any graph  $H$  in a list  $\mathcal{H}$  of nineteen small graphs, via a sequence of problem reductions. In this naming scheme, the prison is the complement of  $H_1$ . The exclusion of  $\text{co-}H_1$  from this list introduces a sequence of new minimal graphs  $H'$  for which the kernelization problem is open, out of which the smallest are the prison plus a vertex  $v$  which is (respectively) an isolated vertex; attached to a degree-3 vertex of the prison; or attached to both a degree-3 and a degree-2 vertex of the prison (R. B. Sandeep, personal communication, using software published along with [15]). It is at the moment not known whether the new list  $\mathcal{H}'$  is finite using the methods of [15].

More broadly, the result suggests that the picture of kernelizability of  $H$ -free Edge Modification problems could be more complex than conjectured by Marx and Sandeep. If so, the question of precisely where the tractability line goes for polynomial kernelization seems highly challenging, as all kernelization results so far (including ours) rely on highly case-specific structural characterizations of  $H$ -free graphs. We leave these deeper investigations into the problem open for future work. We also leave open the question of a polynomial kernel for PRISON-FREE EDGE EDITING.

---

## References

- 1 Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *SIAM J. Discret. Math.*, 28(1):277–305, 2014. doi:10.1137/120880240.
- 2 Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Inf. Process. Lett.*, 58(4):171–176, 1996. doi:10.1016/0020-0190(96)00050-6.
- 3 Leizhen Cai and Yufei Cai. Incompressibility of  $H$ -free edge modification problems. *Algorithmica*, 71(3):731–757, 2015. doi:10.1007/S00453-014-9937-X.
- 4 Yixin Cao, Ashutosh Rai, R. B. Sandeep, and Junjie Ye. A polynomial kernel for diamond-free editing. *Algorithmica*, 84(1):197–215, 2022. doi:10.1007/S00453-021-00891-Y.
- 5 Christophe Crespelle, Pål Grønås Drange, Fedor V. Fomin, and Petr A. Golovach. A survey of parameterized algorithms and the complexity of edge modification. *Comput. Sci. Rev.*, 48:100556, 2023. doi:10.1016/J.COSREV.2023.100556.
- 6 Eduard Eiben and William Lochet. A polynomial kernel for line graph deletion. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPICs*, pages 42:1–42:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.ESA.2020.42.
- 7 Eduard Eiben, William Lochet, and Saket Saurabh. A polynomial kernel for paw-free editing. In *IPEC*, volume 180 of *LIPICs*, pages 10:1–10:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.IPEC.2020.10.
- 8 Paul Erdős and Richard Rado. Intersection theorems for systems of sets. *Journal of the London Mathematical Society*, 1(1):85–90, 1960.
- 9 Michael R. Fellows, Michael A. Langston, Frances A. Rosamond, and Peter Shaw. Efficient parameterized preprocessing for cluster editing. In Erzsébet Csuhaj-Varjú and Zoltán Ésik, editors, *Fundamentals of Computation Theory, 16th International Symposium, FCT 2007, Budapest, Hungary, August 27-30, 2007, Proceedings*, volume 4639 of *Lecture Notes in Computer Science*, pages 312–321. Springer, 2007. doi:10.1007/978-3-540-74240-1\_27.
- 10 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*, volume XIV of *Texts in Theoretical Computer Science. An EATCS Series*. Springer, Berlin, 2006. doi:10.1007/3-540-29953-X.
- 11 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: Theory of Parameterized Preprocessing*. Cambridge University Press, 2019. doi:10.1017/9781107415157.



- 12 Jens Gramm, Jiong Guo, Falk Hüffner, and Rolf Niedermeier. Graph-modeled data clustering: Exact algorithms for clique generation. *Theory Comput. Syst.*, 38(4):373–392, 2005. doi:10.1007/S00224-004-1178-Y.
- 13 Sylvain Guillemot, Frédéric Havet, Christophe Paul, and Anthony Perez. On the (non-)existence of polynomial kernels for  $P_t$ -free edge modification problems. *Algorithmica*, 65(4):900–926, 2013. doi:10.1007/S00453-012-9619-5.
- 14 Stefan Kratsch and Magnus Wahlström. Two edge modification problems without polynomial kernels. *Discret. Optim.*, 10(3):193–199, 2013. doi:10.1016/J.DISOPT.2013.02.001.
- 15 Dániel Marx and R. B. Sandeep. Incompressibility of H-free edge modification problems: Towards a dichotomy. *J. Comput. Syst. Sci.*, 125:25–58, 2022. doi:10.1016/J.JCSS.2021.11.001.
- 16 Stephan Olariu. Paw-fee graphs. *Inf. Process. Lett.*, 28(1):53–54, 1988. doi:10.1016/0020-0190(88)90143-3.
- 17 W. D. Wallis and G. H. Zhang. On maximal clique irreducible graphs. *J. Comb. Math. Comb. Comput*, 8:187–198, 1990.