A Polynomial Kernel for Paw-Free Editing

Eduard Eiben
Department of Computer Science, Royal Holloway, University of London, Egham, UK
eduard.eiben@rhul.ac.uk

William Lochet
Department of Informatics, University of Bergen, Norway
william.lochet@uib.no

Saket Saurabh
Institute of Mathematical Sciences, Chennai, India
Department of Informatics, University of Bergen, Norway
saket@imsc.res.in

Abstract

For a fixed graph H, the H-free Edge Editing problem asks whether we can modify a given graph G by adding or deleting at most k edges such that the resulting graph does not contain H as an induced subgraph. The problem is known to be NP-complete for all fixed H with at least 3 vertices and it admits a $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$ algorithm. Cai and Cai [Algorithmica (2015) 71:731–757] showed that, assuming $\mathsf{coNP} \not\subseteq \mathsf{NP/poly}$, H-free Edge Editing does not admit a polynomial kernel whenever H or its complement is a path or a cycle with at least 4 edges or a 3-connected graph with at least one edge missing. Based on their result, very recently Marx and Sandeep [ESA 2020] conjectured that if H is a graph with at least 5 vertices, then H-free Edge Editing has a polynomial kernel if and only if H is a complete or empty graph, unless $\mathsf{coNP} \subseteq \mathsf{NP/poly}$. Furthermore they gave a list of 9 graphs, each with five vertices, such that if H-free Edge Editing for these graphs does not admit a polynomial kernel, then the conjecture is true. Therefore, resolving the kernelization of H-free Edge Editing for graphs H with 4 and 5 vertices plays a crucial role in obtaining a complete dichotomy for this problem. In this paper, we positively answer the question of compressibility for one of the last two unresolved graphs H on 4 vertices. Namely, we give the first polynomial kernel for Paw-free Edge Editing with $\mathcal{O}\left(k^6\right)$ vertices.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Theory of computation → Graph algorithms analysis; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Kernelization, Paw-free graph, H-free editing, graph modification problem

Digital Object Identifier 10.4230/LIPIcs.IPEC.2020.10

Funding *William Lochet*: Received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 819416).
Saket Saurabh: Received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 819416), and Swarnajayanti Fellowship (No DST/SJF/MSA01/2017-18).

Acknowledgements The authors wish to thank the anonymous referees for helping in the presentation of the paper and pointing out some missing argument in Section 5.

1 Introduction

For a family of graphs $\mathcal{G}$, the general $\mathcal{G}$-Graph Modification problem asks whether we can modify a graph G into a graph in $\mathcal{G}$ by performing at most k simple operations. Typical examples of simple operations that are well-studied in the literature include vertex deletion, edge deletion, edge addition, or combination of edge deletion and addition. We call these problems $\mathcal{G}$-Vertex Deletion, $\mathcal{G}$-Edge Deletion, $\mathcal{G}$-Edge Addition, and

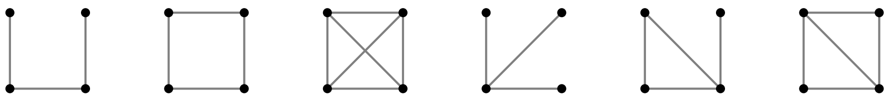
$\mathcal{G}$-Edge Editing, respectively. While a classical result by Lewis and Yannakakis [15] shows that $\mathcal{G}$-Vertex Deletion is NP-complete for all non-trivial hereditary graph classes, the problem seems more difficult for the Edge Modification version and to this day, no simple classification exists.

$\mathcal{G}$-Graph Modification problems have been extensively investigated for graph classes $\mathcal{G}$ that can be characterized by a finite set of forbidden induced subgraphs. We say that a graph is *$\mathcal{H}$-free*, if it does not contain any graph in $\mathcal{H}$ as an induced subgraph. For this special case, the $\mathcal{H}$-free Vertex Deletion problem is well understood. If $\mathcal{H}$ contains a graph on at least two vertices and the class of $\mathcal{H}$-free graphs is non-trivial, then all of these problems are NP-complete, but admit $c^k n^{\mathcal{O}(1)}$ algorithm [3], where c is the size of the largest graph in $\mathcal{H}$ (the algorithms with running time $f(k)n^{\mathcal{O}(1)}$ are called fixed-parameter tractable (FPT) algorithms [9, 11]). Finally, Flum and Grohe [12] showed the existence of a *kernel* with $\mathcal{O}(k^c)$ vertices, where c is again the size of the largest graph in $\mathcal{H}$. A kernel is a polynomial time preprocessing algorithm which outputs an equivalent instance of the same problem such that the size of the reduced instance is bounded by some function $f(k)$ that depends only on k. We call the function $f(k)$ the size of the kernel. It is well-known that any problem that admits an FPT algorithm admits a kernel. Therefore, for problems with FPT algorithms one is interested in polynomial kernels, i.e., kernels with the size upper bounded by a polynomial function of the parameter.

For the edge modification problems, the situation is more complicated. While all of these problems also admit $c^{2k}n^{\mathcal{O}(1)}$ time algorithm, where c is the maximum number of vertices in a graph in $\mathcal{H}$ [3], the P vs NP dichotomy is still not known. Only recently Aravind et al. [1] gave the dichotomy for the special case when $\mathcal{H}$ contains precisely one graph H [1]. From the kernelization point of view, the situation is even more difficult. The reason is that deleting or adding an edge to a graph can introduce a new copy of H and this might further propagate. Hence, we cannot use the sunflower lemma to reduce the size of the instance. Cai asked the question whether H-free Edge Deletion admits a polynomial kernel for all graphs H [2]. Kratsch and Wahlström [14] showed that this is probably not the case and gave a graph H on 7 vertices such that H-free Edge Deletion and H-free Edge Editing do not admit a polynomial kernel unless coNP $\subseteq$ NP/poly. Consequently, it was shown that this is not an exception, but rather a rule [4, 13]. Indeed the result by Cai and Cai [4] shows that H-free Edge Deletion, H-free Edge Addition, and H-free Edge Editing do not admit a polynomial kernel whenever H or its complement is a path or a cycle with at least 4 edges or a 3-connected graph with at least 2 edges missing (resp. at least 1 edge missing in the case of H-free Edge Editing). This suggests that actually the H-free edge modification problems with polynomial kernels are rather rare and only for small graphs H. Based on these observations, very recently Marx and Sandeep [16] conjectured that if H is a graph with at least 5 vertices, then H-free Edge Editing has a polynomial kernel if and only if H is a complete or empty graph, unless coNP $\subseteq$ NP/poly. Furthermore they gave a list of 9 graphs, each with 5 vertices, such that if H-free Edge Editing for all of these graphs does not admit a polynomial kernel, then the conjecture is true. For the graphs on 4 vertices the kernelization of H-free edge modification problems was open for last two graphs and their complements (see Table 1), namely paw and claw, and Cao et al. [7] conjectured that all of these problems admit polynomial kernels. In this paper, we give kernels for the first of the two remaining graphs, namely the paw[1].

---

[1] Independent of our work Cao et al. [6] obtained polynomial kernels for Paw-free Edge Deletion and Paw-free Edge Addition.



(a) P_4 **(b)** C_4 **(c)** K_4 **(d)** claw **(e)** paw **(f)** diamond

Figure 1 List of graphs on 4 vertices, excluding their complements.

Table 1 The kernelization results of H-free edge modification problems for H being 4-vertex graphs. Note that for a complement of H, the rows with deletion and addition are swapped, but otherwise the same results hold.

H	deletion	addition	editing
K_4	$\mathcal{O}\left(k^4\right)$ [5]	trivial	$\mathcal{O}\left(k^4\right)$ [5]
P_4	$\mathcal{O}\left(k^3\right)$ [13]	$\mathcal{O}\left(k^3\right)$ [13]	$\mathcal{O}\left(k^3\right)$ [13]
diamond	$\mathcal{O}\left(k^3\right)$ [18]	trivial	$\mathcal{O}\left(k^8\right)$ [7]
paw	$\mathcal{O}\left(k^4\right)$ [this paper]	$\mathcal{O}\left(k^3\right)$ [this paper]	$\mathcal{O}\left(k^6\right)$ [this paper]
claw	open	open	open
C_4	no [13]	no [13]	no [13]

1.1 Brief Overview of the Algorithm

Our main result is a polynomial kernel for Paw-free Edge Editing. The key to obtain the kernel is a structural theorem by Olariu [17] that states that every connected paw-free graph is either triangle-free or complete multipartite. We start our kernelization algorithm by finding greedily a maximal set of paws $P_1, \dots P_\ell$ such that for any $1 \leq i < j \leq \ell$, P_i and P_j share at most one vertex. This clearly contains at most k paws and hence at most $4k$ vertices. Let us denote the set of these vertices by S. The goal now is to bound the number of vertices in $G - S$. Bounding the number of vertices belonging to the complete multipartite components of $G - S$ is rather simple. We show that every vertex in S is adjacent to at most 1 complete multipartite component and for each multipartite component, we can reduce the size of each part as well as the number of these parts to $\mathcal{O}(k)$. The triangle-free part is trickier. The difficulty comes from the fact that instead of keeping this part of the graph triangle-free, the optimal solution might want to add some edges to make it complete multipartite. However, we argue that there is always an optimal solution that keeps the vertices at distance at least 5 from S in a triangle-free component. This structural claim allows us to look only for solutions which are not too far away from S "in some sense". Moreover, after some preprocessing of the instance, we can also show that the vertices with more than $4k + 6$ neighbors inside the triangle-free components of $G - S$ cannot end up inside a complete multipartite component. It means that we can mark the relevant vertices in triangle-free components as follows. Set $S_0 := S$ and for every $i < 5$, let S_{i+1} be the set obtained by marking for each vertex of S_i, $4k + 6$ neighbors at distance $i + 1$ from S. The size of the set of the marked vertices is then $\mathcal{O}\left(k^6\right)$. Finally, we can remove the vertices of triangle-free components which have not been marked. This is safe because these vertices are either too far from S to belong to a complete multipartite component, or every way to connect these vertices to S uses vertices with more than $4k + 6$ neighbors inside the triangle-free components of $G - S$ that cannot end up in a complete multipartite component of the reduced instance. This gives us the desired kernel.

2 Preliminaries

We assume familiarity with the basic notations and terminologies in graph theory. We refer the reader to the standard book by Diestel [10] for more information. Let us fix a graph G for the sake of this paragraph. We let $|G| = |V(G)|$ and $||G|| = |E(G)|$. For a set of pairs of vertices $A \subseteq \binom{V(G)}{2}$, we denote by $G\Delta A$ the graph whose set of vertices is $V(G)$ and set of edges is the symmetric difference of $E(G)$ and A. For a set of vertices $S \subseteq V(G)$, we denote by $G[S]$ the graph induced by G on S. We let $N_G(v)$ denote the neighborhood of a vertex $v \in V(G)$ and we omit the subscript G if the graph is clear from the context. For a set of vertices S and a vertex $v \in S$, we often refer to $N_G(v) \setminus S$ as the neighborhood of v in $G - S$. A connected component of G is a maximal, w.r.t. inclusion, set of vertices such that $G[C]$ is connected. For the sake of exposition, when speaking about a connected component C we, depended on the context, mean its set of vertices C or the graph $G[C]$ induced on these vertices. Finally, for sets $A, B \subseteq V(G)$, let $E_G(A, B) = \{ab \mid a \in A, b \in B, ab \in E(G)\}$, *i.e.*, the set of edges with one endpoint in A and the other in B. We again omit the subscript G, if the graph is clear from the context.

Parameterized Algorithms and Kernelization. For a detailed illustration of the following facts the reader is referred to [9, 11]. A *parameterized problem* is a language $\Pi \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet; the second component k of instances $(I, k) \in \Sigma^* \times \mathbb{N}$ is called the *parameter*. A parameterized problem Π is *fixed-parameter tractable* if it admits a *fixed-parameter algorithm*, which decides instances (I, k) of Π in time $f(k) \cdot |I|^{\mathcal{O}(1)}$ for some computable function f.

A *kernelization* for a parameterized problem Π is a polynomial-time algorithm that given any instance (I, k) returns an instance (I', k') such that $(I, k) \in \Pi$ if and only if $(I', k') \in \Pi$ and such that $|I'| + k' \leq f(k)$ for some computable function f. The function f is called the *size* of the kernelization, and we have a polynomial kernelization if $f(k)$ is polynomially bounded in k. It is known that a parameterized problem is fixed-parameter tractable if and only if it is decidable and has a kernelization. However, the kernels implied by this fact are usually of superpolynomial size.

A *reduction rule* is an algorithm that takes as input an instance (I, k) of a parameterized problem Π and outputs an instance (I', k') of the same problem. We say that the reduction rule is *safe* if (I, k) is a *yes*-instance if and only if (I', k') is a *yes*-instance. In order to describe our kernelization algorithm, we present a series of reduction rules.

We will need the following result describing the structure of paw-free graphs [17].

▶ **Theorem 1.** *G is a paw-free graph if and only if each connected component of G is triangle-free or complete multipartite.*

To make a clear distinction between these two cases, we will say that a graph is a complete multipartite graph if it contains at least three parts. In particular, it contains a triangle. For an instance (G, k) of Paw-free Edge Editing, we say that A is a *solution* to (G, k) if $|A| \leq k$ and $G\Delta A$ is paw-free.

3 Reduction Rules

From now on (G, k) will be an instance of Paw-free Edge Editing and we assume $k > 0$. Let us first describe two rules which can be safely applied.

▶ **Reduction Rule 1.** *If X is an independent set of $k+3$ vertices with the same neighborhood, remove a vertex $x \in X$ from the graph.*

Proof of Safeness. Suppose (G, k) is an instance of Paw-free Edge Editing and X is an independent set of $k+3$ vertices with the same neighborhood. Let G' be the graph obtained by removing a vertex of X. We need to show that (G', k) has a solution if and only if (G, k) has one. Since G' is an induced subgraph of G, it is clear that if (G, k) has a solution, then so does (G', k). Let A be a solution to (G', k) and assume $G \Delta A$ contains a paw x_1, x_2, x_3, x_4 with x_1, x_2, x_3 being a triangle and x_4 being adjacent to x_3. Because A is a solution to (G', k), it means that one of the x_i must be the vertex x that we removed from G. Moreover, at most two of the other vertices of X belong to the paw, as x is adjacent to at least one vertex in the paw and X is an independent set. If only one other vertex of X belongs to it, consider the other $k+1$ vertices of X which are not in the paw. They all have the same neighborhood in the paw as x, so A must contain for each of them at least one edge with the paw, or we could replace x with this vertex in the paw, which contradicts the fact that A is a solution of (G', k). However, since A is smaller than $k+1$ we reach a contradiction. If two other vertices of X belong to the paw, then it means that $x = x_4$ and these vertices are x_1 and x_2. Moreover it means that the edge $x_1 x_2$ must be edited as X is an independent set. In that case, consider the other k vertices of X which are not in the paw. For every $y \in X \setminus \{x, x_1, x_2\}$, the solution must contain either the edge $y x_3$ or at least one of the nonedges in $\{y x_1, y x_2\}$, but since $|A \setminus \{x_1 x_3\}| < k$, we reach a contradiction. ◀

If Reduction Rule 1 is applicable, then we can easily find an independent set X with at least $k+3$ vertices and the pairwise same neighborhood. This is because there are at most $|V(G)|$ different open neighborhoods of a vertex in G and for each neighborhood, we can simply pass through all vertices $v \in V(G)$ to find all vertices with the given neighborhood. Therefore, we assume from now on that (G, k) is an instance where Reduction Rule 1 cannot be applied.

Following analogous arguments for the case when X induces a complete multipartite graph with at least $k+5$ parts, we also obtain safeness of the following rule. We note that whenever we apply Reduction Rule 2 we will always provide a suitable X and we will not require that G is irreducible w.r.t. this rule. Hence, in particular it is not required to be able to decide the existence of a suitable set X in polynomial time.

▶ **Reduction Rule 2.** *If X is a complete multipartite subgraph with $k+5$ parts having the same neighborhood outside of X, then remove one part of X from the graph.*

Proof of Safeness. Suppose (G, k) is an instance of Paw-free Edge Editing and X is a complete multipartite subgraph with $k+5$ parts having the same neighborhood outside of X. Let P be an arbitrary part of X and let G' be the graph obtained by removing the part P of X. We need to show that (G', k) has a solution if and only if (G, k) has one. Let A be a solution to (G', k) and assume $G \Delta A$ contains a paw x_1, x_2, x_3, x_4 with x_1, x_2, x_3 being a triangle and x_4 being adjacent to x_3. Because A is a solution to (G', k), it means that one of the x_i must belong to P. Moreover, since the vertices in P have exactly the same neighborhood in G and they form an independent set, this paw can contain at most one vertex from P. Let us call x this vertex. Since X consists of $k+5$ parts, it means that there exists $k+1$ parts different from P and without a vertex in this paw. However we know that every vertex in these parts has exactly the same neighborhood as x inside the paw. This means that for every vertex y in these $k+1$ parts, the solution A contains an edge between y and a vertex in $\{x_1, x_2, x_3, x_4\} \setminus \{x\}$ or $G'[\{y, x_1, x_2, x_3, x_4\} \setminus \{x\}]$ is a paw in $G' \Delta A$. Because there are at least $k+1$ parts of X without a vertex in the paw, it follows that either $|A| > k$ or $G' \Delta A$ is not paw-free, a contradiction with A being a solution to (G, k). ◀

Let $\mathcal{H}$ be a maximal set of paws such that any pair share at most one vertex, i.e the paws in $\mathcal{H}$ are edge and non-edge disjoint, and S the set of vertices appearing in $\mathcal{H}$. From now on we will fix the set S. The following observation is immediate from the maximality of $\mathcal{H}$.

▶ **Observation 2.** *For every vertex $v \in S$, the graph $G - (S \setminus \{v\})$ is paw-free.*

We will now introduce two new rules.

▶ **Reduction Rule 3.** *If there is a pair of adjacent vertices $s_1, s_2 \in V(G)$ with $4k + 6$ common neighbors in the triangle-free components of $G - S$, then remove the edge s_1s_2 and set $k := k - 1$.*

We remark here that while for our proof we only need to apply the above reduction rule for all the pairs s_1, s_2 in S, we can safely apply Reduction Rule 3 to all pairs of adjacent vertices. The safeness of Reduction Rule 3 is implied by the following Lemma:

▶ **Lemma 3.** *Suppose Reduction Rule 1 cannot be applied anymore and let s_1, s_2 be two adjacent vertices in G. If there are at least $4k + 6$ vertices belonging to the triangle-free components of $G - S$ adjacent to both s_1 and s_2, then either (G, k) is a* no*-instance, or every solution uses the edge s_1s_2.*

Proof. Suppose there is a solution A not using the edge s_1s_2 and let T be the set of the common neighbors of s_1 and s_2 that are not incident to any edge in A. Because $|A| \leq k$, we know that $|T| \geq 2k + 6$. Since for all $t \in T$, the vertices t, s_1, s_2 induce a triangle in $G\Delta A$, all vertices in T belong to the same complete multipartite component in $G\Delta A$. Moreover, they can only be in two different parts of this component as they belong to the triangle-free components of $G - S$. This means that $k + 3$ of vertices in T belong to the same part of a complete multipartite component of $G\Delta A$. Since vertices in T are not incident to any edge in A, they have the same neighborhood in G. Therefore, Reduction Rule 1 can be applied, which contradicts the assumptions of the lemma. ◀

▶ **Reduction Rule 4.** *If C is a complete multipartite component of $G - S$ and C_1 is a part of C with at least $3k + 3$ vertices, then remove all the edges between the other parts of C and decrease k by the number of edges removed. If this amount is greater than k, answer* no*.*

The fact that Reduction Rule 4 is safe is implied by the following Lemma:

▶ **Lemma 4.** *Suppose Reduction Rule 1 cannot be applied anymore and assume C is a complete multipartite component of $G - S$. If one part of C has at least $3k + 3$ vertices, then either (G, k) is a* no*-instance, or any solution will remove all the edges between the other parts of C.*

Proof. Let C_1 be a part of C of size at least $3k + 3$. Recall that we consider a graph to be a complete multipartite graph only if it contains at least three parts and let s_1, s_2 be two adjacent vertices of $C - C_1$. Let A be a solution to (G, k) which does not use the edge s_1s_2. A is incident to at most $2k$ vertices, so it means that at least $k + 3$ vertices of C_1 are not incident to any edge of A. Moreover, since s_1s_2 is not in A, these $k + 3$ vertices belong to the same part of a complete multipartite component of $G\Delta A$ and thus have the same neighborhood in G. This is a contradiction, as Reduction Rule 1 cannot be applied anymore. ◀

Note also that if Reduction Rules 3 and 4 can be applied, then it is possible to do it in polynomial time. From now on assume that Reduction Rules 1, 3 and 4 can not be applied.

4 Bounding the Complete Multipartite Components

The next two lemmas allow us to bound the number of vertices belonging to complete multipartite components of $G - S$.

▶ **Lemma 5.** *Let C denote a complete multipartite component of $G - S$. If $|C| \geq (3k + 3)(5k + 5)$, then Reduction Rule 2 can be applied in polynomial time.*

Proof. Because Reduction Rule 4 cannot be applied, we have that every part of C contains at most $(3k + 2)$ vertices. Suppose now that C consists of at least $5k + 5$ parts and recall that, by Observation 2, for every vertex $x \in S$ adjacent to C, $G[C \cup \{x\}]$ is paw-free and hence a complete multipartite graph. Therefore, any such vertex x is adjacent to all but at most one part of C. It follows that all but $|S| \leq 4k$ parts of C are adjacent to all vertices in $N(C) \cap S$ and thus at least $5k + 5 - |S|$ parts of C are adjacent to all the vertices of $N(C) \cap S$ and we can find the complete multipartite subgraph X of C induced on these parts in polynomial time by checking the neighborhoods of all vertices in S. Reduction Rule 2 then applies to X and since it simply remove arbitrary part of X, it can be also executed in polynomial time. ◀

▶ **Lemma 6.** *For every $s \in S$, s is adjacent to at most one complete multipartite component of $G - S$.*

Proof. Suppose $s \in S$ is adjacent to two complete multipartite components C and D. Let x be a vertex of C adjacent to s. Since C is a complete multipartite component, it contains at least 3 parts and, in particular, there exist vertices y and z in C such that x, y, z is a triangle. This implies that one of y and z has to be adjacent to s or it would yield a paw without any edge in S which is not possible by definition of $\mathcal{H}$.

Suppose now that y is adjacent to s (the case z is adjacent to s is identical). Now let d be a vertex of D adjacent to s. Because C and D are two different components of $G - S$, d cannot be adjacent to either x or y, which means that s, x, d and y form a paw without any edge or non-edge in S, a contradiction. ◀

The next section is devoted to proving that, if there exists a solution A, then we can assume that any complete multipartite component of $G\Delta A$ only contains vertices at distance 5 from S.

5 Bounding the Diameter of Relevant Vertices

Let A denote an optimal solution and suppose that, among all the optimal solutions, A is chosen so that the sum of the sizes of the multipartite components in $G\Delta A$ is minimized. In this section, C will denote a complete multipartite component of $G\Delta A$, and $C_1, C_2, \ldots, C_r$ the parts of C (see also Figure 2). Furthermore, we will split the vertices of C into *levels* depending on their distance to S. That is we say that a vertex is in the i-th level, if it is at distance i from S in G and we let L_i denote the set of all vertices in the i-th level, i.e., $L_0 = C \cap S$, $L_1 = C \cap N_G(S)$, $L_2 = C \cap N_G(N_G(S)) \setminus L_0$, and so on. For the part C_i and the level L_j, we let $C_{i,j}$ denote the subset of C_i in j-th level. That is for every $i \in [r]$ and every j such that L_j is not empty we let $C_{i,j} = C_i \cap L_j$. Finally, throughout the section for $i \in [r]$ and some level j, we will need to consider the set of all vertices in the j-th level that are not in C_i, we will denote this set $\overline{C_{i,j}}$, i.e., $\overline{C_{i,j}} = \bigcup_{t \neq i} C_{t,j} = L_j \setminus C_i$.

The goal of this section is to show that, because we chose an optimal solution that minimizes the sum of the sizes of the multipartite components in $G\Delta A$, there is no vertex in the j-th level for $j \geq 5$. Let us first show that the result follows easily when L_0 is empty.

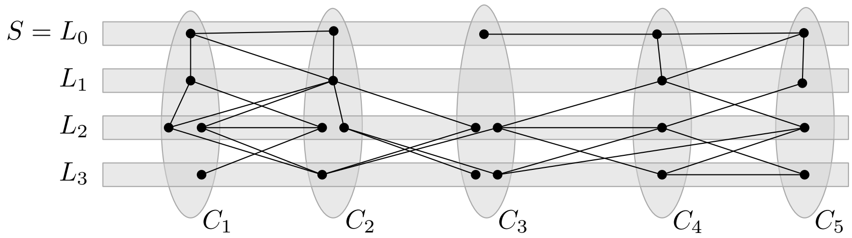


Figure 2 An example of a complete multipartite component C in $G\Delta A$ for some solution A whose vertices were in a triangle-free component of G. The edges drawn are the edges in G. $C_1, \ldots, C_5$ are parts of C, that is, in $G\Delta A$, each C_i is an independent set that is complete to $\bigcup_{j\in[5]\setminus\{i\}} C_j$. L_0, L_1, L_2, and L_3 are the levels in C. That is vertices in L_i are at the distance i from S in G.

▶ **Observation 7.** *If L_0 is empty, then C contains only vertices at distance at most 3 from S.*

Proof. Indeed, if L_0 is empty, then C contains only vertices of $G - S$. In that case, since $G - S$ is paw-free and A is an optimal solution, it follows that A does not contain any pair of vertices of C and thus that C is a complete multipartite component of $G - S$. This ends the proof as the diameter of a complete multipartite graph is 2. ◀

Therefore, from now on we assume that L_0 is not empty. In that case, we observe that it suffices to show that L_5 is empty. Indeed, if some level L_i is empty, then all the edges between the first $i-1$ levels and the remaining levels in $G\Delta A$ are not in G and hence removing them from A gives a smaller solution that splits C into multiple paw-free components. This however contradicts the optimality of A and we get the following observation.

▶ **Observation 8.** *If for some $i \in \mathbb{N}$ is $L_i = \emptyset$, then for all $j > i$ it holds that $L_j = \emptyset$.*

The first step of our proof is the following lemma that basically says that for the set of vertices $C_{i,j}$, the number of edges between $C_{i,j}$ and the rest of C that is added by A has to be smaller than the number of such edges that already exists in G, otherwise we can isolate $C_{i,j}$ from C instead of including it in C and obtain a solution that contradicts our choice of A, because $C_{i,j}$ will not be anymore in a complete multipartite component of the solution.

▶ **Lemma 9.** *For every $j \geq 2$ and every $i \in [r]$ such that $C_{i,j}$ is not empty, if $P \subseteq A$ denotes the set of pairs of A of type xy where $x \in C_{i,j}$ and $y \in \overline{C_{i,j'}}$ with $j' \in \mathbb{N}$, then $|P| < |E_G(C_{i,j}, \overline{C_{i,j-1}} \cup \overline{C_{i,j}} \cup \overline{C_{i,j+1}})|$.*

Proof. See Figure 3 for an illustration. In order to reach a contradiction, suppose this is not the case and consider the set of pairs A' obtained from A by:

- removing all the pairs in P and
- adding $E_G(C_{i,j}, \overline{C_{i,j-1}} \cup \overline{C_{i,j}} \cup \overline{C_{i,j+1}})$.

Because $|P| \geq |E_G(C_{i,j}, \overline{C_{i,j-1}} \cup \overline{C_{i,j}} \cup \overline{C_{i,j+1}})|$, $|A'| \leq |A|$. Moreover, since $A'\Delta A$ are pairs of vertices of C, it means that $(G - C)\Delta A'$ is identical to $(G - C)\Delta A$. We will show now that $G[C]\Delta A'$ consists of one multipartite component $C - C_{i,j}$ and an independent set $C_{i,j}$. Indeed, suppose $x \in C_{i,j}$ and $y \in C \setminus C_{i,j}$, then we can show that yx is not an edge of $G[C]\Delta A'$. The proof can be done by checking the different cases:

- If $y \in C_i$, then the set of pairs of A' containing y is the same as the one in A, and we can conclude since $xy \notin G[C]\Delta A'$.
- If $y \in \overline{C_{i,j'}}$ for some j' such that $|j' - j| > 1$, then because x can only be adjacent to vertices at distance $j, j-1$ and $j+1$, we know that xy is not in $E(G)$ and it does not belong to A', because it is in P.

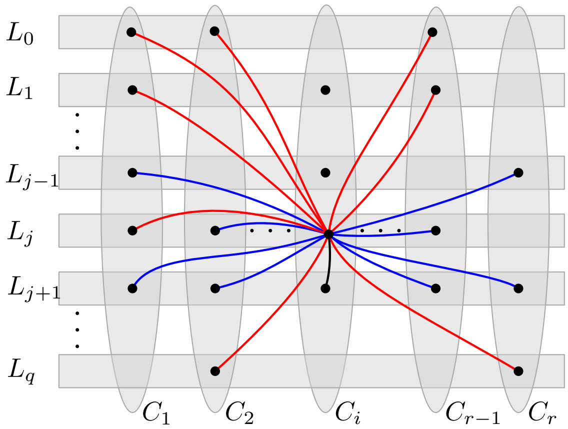


Figure 3 Illustration of Lemma 9. A complete multipartite component C of $G\Delta A$. For simplicity, every nonempty set $C_{s,t}$, $s \in [r]$ and $t \in [q]$, contains only one vertex, but this is not the case in general. The red edges are the set P, *i.e.*, the edges between $C_{i,j}$ and all the parts of C other than C_i added by A. The blue edges are $E_G(C_{i,j}, \overline{C_{i,j-1}} \cup \overline{C_{i,j}} \cup \overline{C_{i,j+1}})$, *i.e.*, all the edges between $C_{i,j}$ and all the parts of C other than C_i that are already in $E(G)$. The black edge is incident to $C_{i,j}$ in G, but its other endpoint is also in C_i, so it is in both A and the solution A' obtained from A by replacing P by $E_G(C_{i,j}, \overline{C_{i,j-1}} \cup \overline{C_{i,j}} \cup \overline{C_{i,j+1}})$ in A.

- If $y \in (\overline{C_{i,j-1}} \cup \overline{C_{i,j}} \cup \overline{C_{i,j+1}})$, then either $xy \in E(G)$ but then the pair belongs to A', or $xy \notin E(G)$ and the pair belongs to A but has been removed in A'.

Overall, A' is a solution to the problem and the complete multipartite components of $G\Delta A'$ are exactly the same as those of $G\Delta A$, except for C which is strictly smaller. This contradicts the choice of A, as $|A'| \leq |A|$. ◀

▶ **Lemma 10.** *If for some $i \in [r]$ the set $C_{i,0} \cup C_{i,1}$ is not empty, then $C_{i,j} = \emptyset$ for every $j \geq 4$.*

Proof. Suppose $C_{i,0} \cup C_{i,1}$ and $C_{i,j}$ are not empty. Because $j \geq 4$, we know that $E_G(C_{i,j}, \overline{C_{i,0}} \cup \overline{C_{i,1}} \cup \overline{C_{i,2}})$ is empty. This implies that A contains all the pairs in $C_{i,j} \times (\overline{C_{i,0}} \cup \overline{C_{i,1}} \cup \overline{C_{i,2}})$. By applying Lemma 9 to $C_{i,j}$, we deduce that

$$|C_{i,j}| \times |\overline{C_{i,0}} \cup \overline{C_{i,1}} \cup \overline{C_{i,2}}| < |E_G(C_{i,j}, \overline{C_{i,j-1}} \cup \overline{C_{i,j}} \cup \overline{C_{i,j+1}})|.$$

However,

$$|E_G(C_{i,j}, \overline{C_{i,j-1}} \cup \overline{C_{i,j}} \cup \overline{C_{i,j+1}})| \leq |C_{i,j}| \times |\overline{C_{i,j-1}} \cup \overline{C_{i,j}} \cup \overline{C_{i,j+1}}|$$

and by combining the two inequalities we obtain that

$$|\overline{C_{i,j-1}} \cup \overline{C_{i,j}} \cup \overline{C_{i,j+1}}| > |\overline{C_{i,0}} \cup \overline{C_{i,1}} \cup \overline{C_{i,2}}|. \tag{1}$$

Consider the set of pairs A', obtained from A by:
- adding $E_G(C_{i,0} \cup C_{i,1}, \overline{C_{i,0}} \cup \overline{C_{i,1}} \cup \overline{C_{i,2}})$ and
- removing all the pairs of the form xy with $x \in C_{i,0} \cup C_{i,1}$ and $y \in C_s$ for $s \neq i$.

By a very similar argument to the one of Lemma 9, we can show that A' is a solution such that $G\Delta A'$ differs from $G\Delta A$ only in the fact that $C_{i,0} \cup C_{i,1}$ has been disconnected from C. Moreover, we know that the set of pairs of the form xy with $x \in C_{i,0} \cup C_{i,1}$ and $y \in C_s$ for $s \neq i$ contains $\big((C_{i,0} \cup C_{i,1}) \times (\overline{C_{i,j-1}} \cup \overline{C_{i,j}} \cup \overline{C_{i,j+1}})\big)$. However, from (1), we can deduce that $|E_G(C_{i,0} \cup C_{i,1}, \overline{C_{i,0}} \cup \overline{C_{i,1}} \cup \overline{C_{i,2}})| < |\big((C_{i,0} \cup C_{i,1}) \times (\overline{C_{i,j-1}} \cup \overline{C_{i,j}} \cup \overline{C_{i,j+1}})\big)|$ and thus $|A'| < |A|$, which gives us a contradiction. ◀

The main implication of Lemma 10 is that, if L_j is not empty for $j \geq 4$, then A contains all the pairs $L_j \times (L_0 \cup L_1)$. Indeed, it shows that vertices in L_j and $L_0 \cup L_1$ belong to different parts and thus must be adjacent in $G \Delta A$. However, just by considering the distance to S in G, these vertices cannot be adjacent in G, and thus these pairs must be in A. This allows us to prove the following lemma.

▶ **Lemma 11.** *For every $j \geq 5$, L_j is empty.*

Proof. First note that by Observation 8, it is enough to show that, for some $j \leq 5$, the set L_j empty. Hence, for the sake of a contradiction, suppose none of $L_0, L_1, \dots, L_5$ is empty. Now by Lemma 10, we know that the vertices in L_5 and $L_0 \cup L_1$ belong to different parts of the complete multipartite component. This implies that A contains $L_5 \times (L_0 \cup L_1)$. Consider A' the set of pairs obtained from A by:

- removing all the pairs $xy \in A$ where $xy \notin E(G)$, $x \in L_5$ and $y \in L_4$,
- adding all the edges of $E_G(L_5, L_4)$ which are not already in A, and
- removing all pairs xy with $x \in L_s$ and $y \in L_f$ for $s \leq 3$ and $f \geq 5$.

By doing so, we only disconnect $\bigcup_{s \leq 4} L_s$ from $\bigcup_{f \geq 5} L_f$ in $G \Delta A'$ compared to $G \Delta A$. This means that A' is also a solution, and by minimality of A, we have that $|A'| \geq |A|$. We can then deduce that $|L_4| \cdot |L_5| \geq E_G(L_5, L_4) \geq |L_5| \cdot |L_0 \cup L_1|$ and thus $|L_4| \geq |L_0 \cup L_1|$.

Now again by Lemma 10, we have that A contains $L_4 \times (L_0 \cup L_1)$. However, $|L_4| \geq |L_0 \cup L_1|$ so it means that $|L_0 \cup L_1|^2 \leq |L_4| \cdot |L_0 \cup L_1|$. Let A'' be the solution obtained from A by:

- removing all the pairs $xy \in A$ where $xy \notin E(G)$, $x \in L_0$ and $y \in L_1$,
- adding all the edges of $E_G(L_0, L_1)$ which are not already in A,
- removing all the pairs $xy \in A$ where $x \in L_0$ and $y \in L_i$ with $i \geq 2$, and
- removing all the pairs $xy \in A$ where $x, y \in C \setminus L_0$.

Note first that again $(G - C)\Delta A$ and $(G - C)\Delta A''$ are the same. Now consider a component D of $G[C]\Delta A''$. Since we removed all the edges between L_0 and the rest of C, D is either subset of S or a subset of $V(G) \setminus S$. In the case that D is a subset of S, then $G[D]\Delta A'' = G[D]\Delta A$ and $G[D]\Delta A''$ is an induced subgraph of the complete multipartite graph $G[C]\Delta A$ and hence either complete multipartite or triangle-free. In the case that D is a subset of $V(G) \setminus S$, then we removed from A all the pair that have one endpoint in D and the other anywhere in C (including D), so $G[D]\Delta A''$ is an induced subgraph of exactly one connected component of $G - S$. Since all components of $G - S$ are paw-free and being paw-free is a property closed under taking induced subgraphs, it follows that $G[D]\Delta A''$ is also paw-free. Therefore, we conclude that $G\Delta A''$ is paw-free and A''. It remains to show that A'' contradicts the choice of A.

The set $A'' \setminus A$ contains only edges in $E_G(L_0, L_1)$, so $|A'' \setminus A| \leq |L_0 \cup L_1|^2 \leq |L_4| \cdot |L_0 \cup L_1|$. On the other hand, by Lemma 10, the fact that G does not contain edges between L_i and L_j for $|i - j| > 1$, and by the construction of A'', we have that $A \setminus A''$ contains all the pairs xy such that $x \in L_0 \cup L_1$ and $y \in L_4 \cup L_5$. In particular, $|A \setminus A''| \geq |L_0 \cup L_1| \cdot |L_4 \cup L_5|$, but since L_5 is not empty, we have

$$|A \setminus A''| \geq |L_0 \cup L_1| \cdot |L_4 \cup L_5| > |L_0 \cup L_1| \cdot |L_4| \geq |A'' \setminus A|,$$

and it follows that $|A''| < |A|$, which contradicts the optimality of A. ◀

6 Triangle-Free Components

Before proving our main result let us prove the following lemma, which will be useful in bounding the number of vertices outside of S.

▶ **Lemma 12.** *If $x \in G$ has at least $4k+6$ neighbors belonging to triangle-free components of $G - S$, then there is no solution A such that x belongs to a complete multipartite component of $G\Delta A$.*

Proof. Let T denote the set of neighbors of x belonging to triangle-free components of $G-S$. Suppose x belongs to a complete multipartite component C of $G\Delta A$. First note that at least $2k+6$ of the vertices of T will not be adjacent to any edge of A, which means that their neighborhood in G and $G\Delta A$ are the same and they belong to C in $G\Delta A$. Now because the vertices of T belong to triangle-free components, it means that these $2k+6$ vertices can only belong to two different parts of this multipartite component. In particular, at least $k+3$ of those belong to the same part and thus have the exact same neighborhood in $G\Delta A$ and thus in G. This means that Reduction Rule 1 can be applied, which is a contradiction. ◀

▶ **Lemma 13.** *Suppose (G, k) is a* yes*-instance. Then there exists a set S' of at most $(4k+6)4k$ vertices such that if $x \notin S'$ belongs to a triangle-free component of $G - S$, then x does not belong to any triangle in G using only one vertex of $S \cup S'$. Moreover, there is a polynomial time algorithm that either finds this set or concludes that (G, k) is a* no*-instance.*

Proof. Let x be a vertex belonging to a triangle-free component C of $G - S$. Suppose that x belongs to a triangle using only one vertex s of S and another vertex y of C. Note first that C is the only component of $G - S$ adjacent to s or we would have a paw with only one vertex in S (which is impossible by Observation 2). Suppose now that $t \in C$ is adjacent to x. Then t must be adjacent to either y or s or it would yield a paw with only one vertex in S. Thus, since C is triangle free, t must be adjacent to s. The same argument would show that any vertex adjacent to t in C must be adjacent to s and thus the whole component C is adjacent to s (by symmetry of x and y).

Now let $s \in S$ and let C_s denote a triangle-free component of $G - S$ such that there exist two vertices $x, y \in C_s$ that induce a triangle with s. Note that if such a component exists, then, by the above argument, it is the unique component in $G-S$ adjacent to s, more precisely $C_s = N(s) \cap (V(G - S))$, and let us consider only the vertices in S for which such a component exists.

Let $\mathcal{M}_s$ be a maximal matching in C_s. If $\mathcal{M}_s$ consists of more than k edges, then it means that any solution A to the instance (G, k) puts s in a complete multipartite component. In particular if $|C_s| \geq 4k+6$, as $C_s \subseteq N(s)$ and $|A| \geq k$, we have that $2k+6$ of the vertices of C_s are not adjacent to any edge of A and belong to the same complete multipartite component of $G\Delta A$ as s. Moreover, these $2k + 6$ vertices can only belong to two different parts of this complete multipartite component (or we would have a triangle in C_s), and thus $k+3$ of them belong to the same part. However, since their neighborhood in G and $G\Delta A$ are identical, it means we could have applied Reduction Rule 1. Hence, if $|C_s| \geq 4k + 6$, then the solution A cannot exist and we can conclude that (G, k) is a *no*-instance. Otherwise, let C'_s be the set of the vertices of $\mathcal{M}_s$ and note that the vertices in $C_s \setminus C'_s$ induce an independent set in G. In particular, the vertices in $C_s \setminus C'_s$ are singletons in $G - (S \cup C'_s)$. and hence no vertex in $C_s \setminus C'_s$ forms a triangle with another vertex in $G - (S \cup C'_s)$.

Let $S' = \bigcup_{s \in S} C'_s$, where $C'_s = \emptyset$ if the component C_s does not exists, *i.e.*, if there is no triangle in G containing s and two vertices in a triangle-free component. By the construction of C'_s for each $s \in S$, it follows that no vertex x in a triangle-free component of $V(G) \setminus (S \cup S')$ belongs to a triangle using only one vertex of $S \cup S'$. Moreover, either $|S'| \leq |S| \cdot (4k+6) \leq (4k+6)4k$, or there is $s \in S$ such that $|\mathcal{M}_s| > k$ and $|C_s| \geq 4k+6$ and we can conclude that (G,k) is *no*-instance. ◀

7 Main Result

▶ **Theorem 14.** *Paw-free Edge Editing has a kernel on* $\mathcal{O}\left(k^6\right)$ *vertices.*

Proof. To ensure that the reduction rules are applied in the correct order, that is, *e.g.*, that we never apply Reduction Rule 3 if Reduction Rule 1 can be applied, we restart the algorithm from the beginning on the reduced instance whenever it is reduced according to some reduction rule. Since every reduction rule decreases either number of vertices of G or the parameter, this increases the running time at most by the factor of $|G| + k$.

Let (G,k) be an instance of Paw-free Edge Editing. The algorithm first applies Reduction Rule 1. If Reduction Rule 1 cannot be applied anymore, the algorithm computes $\mathcal{H}$ a maximal packing of edge-disjoint paws. If $\mathcal{H}$ consists of more than k paws, answer *no*. If this is not the case, let S be the set of vertices belonging to a paw of $\mathcal{H}$. As S is the union of at most k paws, $|S| \leq 4k$. Then the algorithm applies Reduction Rules 3 and 4 until either $k < 0$, in which case it answers *no*, or they cannot be applied anymore.

Because $\mathcal{H}$ is maximal, Theorem 1 implies that the components $G - S$ are either triangle-free or complete multipartite. Let C be a complete multipartite component. If $|C| \geq (3k+3)(5k+5)$, then Lemma 5 implies that the algorithm can apply Reduction Rule 2. Moreover Lemma 6 implies that the number of complete multipartite components adjacent to S is bounded by $|S|$. Overall this implies that the number of vertices contained in complete multipartite components of $G - S$ adjacent to S is bounded by $4k(3k+3)(3k+5)$, or it is possible to apply Reduction Rule 2.

By applying Lemma 13, we either find out that (G,k) is a *no*-instance or find a set S' of at most $(4k+6)4k$ vertices such that if $x \notin S'$ belongs to a triangle-free component of $G - S$, then x does not belong to any triangle in G using only one vertex of S.

Because Reduction Rule 3 cannot be applied anymore, it means that for every pair of adjacent vertices s_1, s_2 in S, the number of vertices in triangle-free components adjacent to both s_1 and s_2 is bounded by $4k+6$. This means that, if S'' denotes the set of vertices in a triangle-free component forming a triangle with 2 vertices of S, then $|S''| \leq |S|^2(4k+6)$.

Then we construct recursively sets $S_0, S_1, \ldots, S_6$ such that S_i is a subset of vertices of G at distance i from S as follows: We set $S_0 := S$. Now we proceed in 6 rounds. In the i-th round we mark, for every vertex $x \in S_{i-1}$, arbitrary $4k+6$ neighbors of x at distance i from S in G and belonging to a triangle-free component of $G - S$. Afterwards, we let S_i be the set of vertices marked in this round and proceed to the next round. Note that $|\bigcup S_i| = \mathcal{O}\left(k^6\right)$.

Let G' be the graph induced on G by S, S', S'', all the sets S_i for $i \in [6]$ and all the complete multipartite components of $G - S$ adjancent to S. Note that, by construction of S' and S'', there is no triangle in G using a vertex which is not in G'. We claim that (G',k) has a solution if and only if (G,k) has a solution. As G' is a subgraph of G, it is clear that if (G,k) has a solution, then so does (G',k). Suppose now that (G',k) has a solution A, but (G,k) does not have a solution. In particular, it implies that $G \Delta A$ is not paw-free. Because of Lemma 11, we can assume that no complete multipartite component of $G' \Delta A$ has a vertex at distance 5 from S and that A is minimal. Let x_1, x_2, x_3, x_4 form a paw in $G \Delta A$, with x_1, x_2, x_3 being the triangle. If x_1, x_2, x_3 is a triangle of $G'\Delta A$, then x_4 is a vertex of $G - G'$ adjacent to one of these vertices, say x_1. Since x_1 is at distance less than 5 from S, it means that during the marking process x_4 was not marked for x_1 and that x_1 has more than $4k+6$ neighbors in triangle-free components of $G' - S$. However, Lemma 12 implies that x_1 cannot belong to a complete multipartite component of $G'\Delta A$, which is a contradiction. If x_1, x_2, x_3 is not a triangle of $G'\Delta A$, then, without loss of generality, we can assume that x_1 belongs to $G - G'$, x_2 and x_3 belong to G', and the edge x_2x_3 was added by A. If x_2 and x_3 belong to a triangle-free component of $G'\Delta A$, then $A \setminus \{x_2, x_3\}$ is a smaller solution to (G', k). Therefore, x_2 and x_3 belong to a complete multipartite component of $G'\Delta A$. This means that they are at distance at most 5 from S and x_1 was not marked for both x_2 and x_3 during the marking process. Finally, this implies that both x_2 and x_3 already have more than $4k+6$ neighbors in triangle-free components of $G' - S$ and thus cannot belong to a multipartite component of $G' - S$, a contradiction. ◀

8 Kernels for Deletion and Addition

In this section, we provide kernels for Paw-free Edge Deletion and Paw-free Edge Addition. To obtain the kernel for these problems, we observe that Reduction Rules 1–4 apply even if we are only allowed to delete respectively only allowed to add edges. This allows us to reduce the complete multipartite components. Furthermore, by deleting the edges, we cannot change a triangle-free component to a complete multipartite one and it actually suffice to keep the vertices that actually appear in triangle together with $4k+6$ of each of such vertex, which can be bounded using Lemma 13. For the edge addition, we just observe that every connected component of G that contains a paw, and hence a triangle, has to be modified to a complete multipartite graph and we can basically conclude by Lemma 12.

▶ **Theorem 15.** *Paw-free Edge Deletion admits a kernel with $\mathcal{O}\left(k^4\right)$ vertices.*

Proof. Let (G, k) be an instance of Paw-free Edge Deletion. First note that Reduction Rules 1–4 are still safe in this context, and Lemma 12 still applies. Therefore the algorithm applies Reduction Rule 1 until it cannot be applied anymore. It then computes $\mathcal{H}$ a maximal packing of edge-disjoint paws. If $\mathcal{H}$ consists of more than k paws, answer *no*. If this is not the case, let S be the set of vertices belonging to a paw of $\mathcal{H}$. As S is the union of at most k paws, $|S| \leq 4k$. Then the algorithm apply Reduction Rules 3 and 4 until either $k < 0$, in which case it answers *no*, or they cannot be applied anymore.

Again, by possibly applying Reduction Rule 2, we can assume that the set of vertices in all the multipartite components of $G - S$ adjacent to S is smaller than $4k(3k+3)(3k+5)$. By applying Lemma 13, we either find out that (G, k) is a *no*-instance or find a set S' of at most $(4k+6)4k$ vertices such that if $x \notin S'$ belongs to a triangle-free component of $G - S$, then x does not belong to any triangle in G using only one vertex of S.

Because Reduction Rule 3 cannot be applied anymore, it means that for every pair of adjacent vertices s_1, s_2 in S, the number of vertices in triangle-free components adjacent to both s_1 and s_2 is bounded by $4k+6$. This means that, if S'' denote the set of vertices in a triangle-free component, forming a triangle with 2 vertices of S, then $|S''| \leq |S|^2(4k+6)$.

Note also that Lemma 12 still applies, and let S_1 be the set obtained by picking for every vertex s in $S \cup S' \cup S''$, $4k+6$ neighbors in triangle-free components of $G - S$.

Let G' be the graph induced on G by S, S', S'', S_1, as well as all the vertices on complete multipartite components of $G - S$. We want to show that (G, k) has a solution if and only if (G', k) has a solution. Let A be a solution of (G', k) and suppose $G\Delta A$ has a paw

x_1, x_2, x_3, x_4, with x_1, x_2, x_3 being a triangle and x_4 being adjacent to x_3. Because of the choice of the sets S' and S'', all the triangles of G are contained in G'. Note also that, since the solution can only remove edges, x_1, x_2, x_3 is a triangle in G. In particular, x_1, x_2, x_3 is a triangle in G' and $x_4 \notin V(G')$. This implies that $x_3 \in S \cup S' \cup S''$ and x_4 was not picked for the $4k+6$ neighbors of x_3. In particular, this means that x_3 has $4k+6$ neighbors which belong to a triangle-free component of $G' - S$ in G' and thus, by Lemma 12, x_3 cannot belong to a complete multipartite component of $G' \Delta A$. However, since x_1, x_2 and x_3 form a triangle in $G' \Delta A$, we reach a contradiction. ◀

▶ **Theorem 16.** *Paw-free Edge Addition admits a kernel with $\mathcal{O}\left(k^3\right)$ vertices.*

Proof. Again, Reduction Rules 1–4 are still safe in this context, with the difference for Rules 3 and 4 that, instead of removing edges and decreasing k, we can directly conclude that (G, k) is a *no*-instance. Note also that a paw-free connected component can safely be removed from the graph.

So the algorithm starts by removing all the paw-free components of G and applying Reduction Rule 1 until it cannot be applied anymore. It then computes $\mathcal{H}$ a maximal packing of edge-disjoint paws. If $\mathcal{H}$ consists of more than k paws, answer *no*. If this is not the case, let S be the set of vertices belonging to a paw of $\mathcal{H}$. As S is the union of at most k paws, $|S| \leq 4k$. From now on we can assume that Rules 3 and 4 cannot be applied.

Again, by possibly applying Reduction Rule 2, we can assume that the set of vertices in all the multipartite components of $G - S$ adjacent to S is smaller than $4k(3k+3)(3k+5)$.

Consider a connected component C_1 of G. This component cannot be paw-free, or the algorithm would have removed it from the graph. So let $S_1 = C_1 \cap S$ and R_1 the vertices of C_1 contained in triangle-free component of $G - S$. Because C_1 is not triangle-free, it means that any solution A to (G, k) leaves C_1 as a complete multipartite component. In particular, it implies that R_1 is smaller than $4k+6$. Indeed, if R_1 is bigger than $4k+6$, then $2k+6$ vertices will have the same neighborhood in $G \Delta A$ as in G. Moreover, since R_1 is triangle-free, it means that these vertices belong to at most 2 parts of the complete multipartite component. This implies that at least $k+3$ of these vertices belong to the same part and Rule 1 applies. Moreover, since G has at most k connected components which are not paw-free, it implies that the set of vertices contained in triangle-free components of $G - S$ is smaller than $(4k+6)k$.

Overall, it implies that our reduced instance has size at most $4k(3k+3)(3k+5) + (4k+6)k + 4k = \mathcal{O}\left(k^3\right)$, which ends the proof. ◀

9 Conclusion

In this paper we studied Paw-free Edge Editing and gave a polynomial kernel of size $\mathcal{O}\left(k^6\right)$. The only unresolved graph H on 4 vertices, for which the kernelization complexity of H-free Edge Editing problem remains open is the claw. In fact, for this problem even the kernelization complexity of H-Edge Deletion and H-Edge Addition remain open. Settling the kernelization complexity might require using the power of structure theorem of claw free graphs [8]. Thus, a natural start here could be looking at editing/deletion/addition to basic graphs, on which structure theorem of claw free graphs is built. We leave these as natural directions to pursue.

References

1 N. R. Aravind, R. B. Sandeep, and Naveen Sivadasan. Dichotomy results on the hardness of H-free edge modification problems. *SIAM J. Discrete Math.*, 31(1):542–561, 2017. doi:10.1137/16M1055797.

2 Hans L Bodlaender, Leizhen Cai, Jianer Chen, Michael R Fellows, Jan Arne Telle, and Dániel Marx. Open problems in parameterized and exact computation-iwpec 2006. *UU-CS*, 2006, 2006.

3 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.

4 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.

5 Yufei Cai. Polynomial kernelisation of H-free edge modification problems. Mphil thesis, Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China, 2012.

6 Yixin Cao, Yuping Ke, and Hanchun Yuan. Polynomial kernels for paw-free edge modification problems. *CoRR*, abs/2003.11273, 2020. arXiv:2003.11273.

7 Yixin Cao, Ashutosh Rai, R. B. Sandeep, and Junjie Ye. A polynomial kernel for diamond-free editing. In *26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland*, pages 10:1–10:13, 2018. doi:10.4230/LIPIcs.ESA.2018.10.

8 Maria Chudnovsky and Paul D. Seymour. Claw-free graphs. IV. decomposition theorem. *J. Comb. Theory, Ser. B*, 98(5):839–938, 2008. doi:10.1016/j.jctb.2007.06.007.

9 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

10 R. Diestel. *Graph Theory, 4th Edition*. Springer, 2012.

11 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

12 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. doi:10.1007/3-540-29953-X.

13 Sylvain Guillemot, Frédéric Havet, Christophe Paul, and Anthony Perez. On the (non-)existence of polynomial kernels for P_ℓ-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. *Discrete Optimization*, 10(3):193–199, 2013. doi:10.1016/j.disopt.2013.02.001.

15 John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *J. Comput. Syst. Sci.*, 20(2):219–230, 1980. doi:10.1016/0022-0000(80)90060-4.

16 Dániel Marx and R. B. Sandeep. Incompressibility of H-Free Edge Modification Problems: Towards a Dichotomy. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms (ESA 2020)*, volume 173 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 72:1–72:25, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ESA.2020.72.

17 Stephan Olariu. Paw-free graphs. *Information Processing Letters*, 28(1):53–54, 1988. doi:10.1016/0020-0190(88)90143-3.

18 R. B. Sandeep and Naveen Sivadasan. Parameterized lower bound and improved kernel for diamond-free edge deletion. In *10th International Symposium on Parameterized and Exact Computation, IPEC 2015, September 16-18, 2015, Patras, Greece*, pages 365–376, 2015. doi:10.4230/LIPIcs.IPEC.2015.365.