On the Parameterized Approximability of Contraction to Classes of Chordal Graphs[*]

SPOORTHY GUNDA, Simon Fraser University, Burnaby, Canada.

PALLAVI JAIN, Indian Institute of Technology Jodhpur, Jodhpur, India.

DANIEL LOKSHTANOV, University of California, Santa Barbara, USA.

SAKET SAURABH, The Institute Of Mathematical Sciences, HBNI, Chennai, India and University of Bergen, Bergen, Norway

PRAFULLKUMAR TALE, Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

A graph operation that *contracts edges* is one of the fundamental operations in the theory of graph minors. Parameterized Complexity of editing to a family of graphs by contracting k edges has recently gained substantial scientific attention, and several new results have been obtained. Some important families of graphs, namely the subfamilies of chordal graphs, in the context of edge contractions, have proven to be significantly difficult than one might expect. In this paper, we study the $\mathcal{F}$-CONTRACTION problem, where $\mathcal{F}$ is a subfamily of chordal graphs, in the realm of parameterized approximation. Formally, given a graph G and an integer k, $\mathcal{F}$-CONTRACTION asks whether there exists $X \subseteq E(G)$ such that $G/X \in \mathcal{F}$ and $|X| \leq k$. Here, G/X is the graph obtained from G by contracting edges in X. We obtain the following results for the $\mathcal{F}$-CONTRACTION problem.

- CLIQUE CONTRACTION is known to be FPT. However, unless NP $\subseteq$ coNP/*poly*, it does not admit a polynomial kernel. We show that it admits a polynomial-size approximate kernelization scheme (PSAKS). That is, it admits a $(1 + \epsilon)$-approximate kernel with $O(k^{f(\epsilon)})$ vertices for every $\epsilon > 0$.
- SPLIT CONTRACTION is known to be W[1]-Hard. We deconstruct this intractability result in two ways. Firstly, we give a $(2 + \epsilon)$-approximate polynomial kernel for SPLIT CONTRACTION (which also implies a factor $(2 + \epsilon)$-FPT-approximation algorithm for SPLIT CONTRACTION). Furthermore, we show that, assuming Gap-ETH, there is no $\left(\frac{5}{4} - \delta\right)$-FPT-approximation algorithm for SPLIT CONTRACTION. Here, $\epsilon, \delta > 0$ are fixed constants.
- CHORDAL CONTRACTION is known to be W[2]-Hard. We complement this result by observing that the existing W[2]-hardness reduction can be adapted to show that, assuming FPT$\neq$ W[1], there is no $F(k)$-FPT-approximation algorithm for CHORDAL CONTRACTION. Here, $F(k)$ is an arbitrary function depending on k alone.

Authors' addresses: Spoorthy Gunda, Simon Fraser University, Burnaby, Canada., sgunda@sfu.ca; Pallavi Jain, Indian Institute of Technology Jodhpur, Jodhpur, India., pallavi@iitj.ac.in; Daniel Lokshtanov, University of California, Santa Barbara, USA., daniello@ucsb.edu; Saket Saurabh, The Institute Of Mathematical Sciences, HBNI, Chennai, India and University of Bergen, Bergen, Norway, saket@imsc.res.in; Prafullkumar Tale, Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany, prafullkumar.tale@mpi-inf.mpg.de.

111

We say that an algorithm is an $h(k)$-FPT-approximation algorithm for the $\mathcal{F}$-Contraction problem, if it runs in FPT time, and on any input (G, k) such that there exists $X \subseteq E(G)$ satisfying $G/X \in \mathcal{F}$ and $|X| \leq k$, it outputs an edge set Y of size at most $h(k) \cdot k$ for which G/Y is in $\mathcal{F}$.

CCS Concepts: • **Theory of computation → Parameterized complexity and exact algorithms**; *Approximation algorithms analysis*.

Additional Key Words and Phrases: Graph Contraction, FPT-Approximation, Inapproximability, Lossy Kernels

1 INTRODUCTION

Graph modification problems have been extensively studied since the inception of Parameterized Complexity in the early '90s. The input of a typical graph modification problem consists of a graph G and a positive integer k, and the objective is to edit k vertices (or edges) so that the resulting graph belongs to some particular family, $\mathcal{F}$, of graphs. These problems are not only mathematically and structurally challenging, but have also led to the discovery of several important techniques in the field of Parameterized Complexity. It would be completely appropriate to say that solutions to these problems played a central role in the growth of the field. In fact, just in the last few years, parameterized algorithms have been developed for several graph editing problems [5, 6, 9–12, 18–20, 25, 27, 28]. The focus of all of these papers and the vast majority of papers on parameterized graph editing problems has so far been limited to edit operations that delete vertices, delete edges or add edges.

In recent years, a different edit operation has begun to attract significant scientific attention. This operation, which is arguably the most natural edit operation apart from deletions/insertions of vertices/edges, is the one that contracts an edge. Here, given an edge uv that exists in the input graph, we remove the edge from the graph and merge its two endpoints. Edge contraction is a fundamental operation in the theory of graph minors. For some particular family of graphs, $\mathcal{F}$, we say that a graph G belongs to $\mathcal{F} + kv$, $\mathcal{F} + ke$ or $\mathcal{F} - ke$ if some graph in $\mathcal{F}$ can be obtained by deleting at most k vertices from G, deleting at most k edges from G or adding at most k edges to G, respectively. Using this terminology, we say that a graph G belongs to $\mathcal{F}|ke$ if some graph in $\mathcal{F}$ can be obtained by contracting at most k edges in G. In this paper, we study the following problem.

$\mathcal{F}$-Contraction	**Parameter:** k
Input: A graph G and an integer k	
Question: Does G belong to $\mathcal{F}|ke$?	

For several families of graphs $\mathcal{F}$, early papers by Watanabe et al. [47, 48], and Asano and Hirata [3] showed that $\mathcal{F}$-Edge Contraction is NP-complete.

In the framework of Parameterized Complexity, these problems exhibit properties that are quite different from those problems where we only delete or add vertices and edges. Indeed, a well-known result by Cai [7] states that in case $\mathcal{F}$ is a hereditary family of graphs with a finite set of forbidden induced subgraphs, then the graph modification problems, $\mathcal{F} + kv$, $\mathcal{F} + ke$ or $\mathcal{F} - ke$, defined by $\mathcal{F}$ admit a simple FPT algorithm (an algorithm with running time $f(k)n^{O(1)}$). However, for $\mathcal{F}$-Contraction, the result by Cai [7] does not hold. In particular, Lokshtanov et al. [38] and Cai and Guo [8] independently showed that if $\mathcal{F}$ is either the family of P_ℓ-free graphs for some $\ell \geq 5$ or the family of C_ℓ-free graphs for some $\ell \geq 4$, then $\mathcal{F}$-Contraction is W[2]-Hard (W[i]-hardness, for $i \geq 1$, is an analogue to NP-hardness in Parameterized Complexity, and is used to rule out FPT-algorithm for the problem) when parameterized by k (the number of edges to be contracted).

These results immediately imply that CHORDAL CONTRACTION is W[2]-Hard when parameterized by k. The parameterized hardness result for CHORDAL CONTRACTION led to finding subfamilies of chordal graphs, where the problem could be shown to be FPT. Two subfamilies that have been considered in the literature are families of *split graphs* and *cliques*. Cai and Guo [8] showed that CLIQUE CONTRACTION is FPT, however, it does not admit a polynomial kernel. Later, Cai and Guo [32] also claimed to design an algorithm that solves SPLIT CONTRACTION in time $2^{O(k^2)} \cdot n^{O(1)}$, which proves that the problem is FPT. However, Agrawal et al. [2] found an error with the proof and showed that SPLIT CONTRACTION is W[1]-Hard.

> Inspired by the intractable results that CHORDAL CONTRACTION, SPLIT CONTRACTION and CLIQUE CONTRACTION are W[2]-Hard, W[1]-Hard, and does not admit polynomial kernel, respectively, we study them from the viewpoint of parameterized approximation.

Our Results and Methods. We start by defining a few basic definitions in parameterized approximation. To formally define these, we need a notion of parameterized optimization problems. We defer formal definitions to Section 2 and give intuitive definitions here. We say that an algorithm is an $h(k)$-FPT-approximation algorithm for the $\mathcal{F}$-CONTRACTION problem, if it runs in FPT time, and on any input (G, k) if there exists $X \subseteq E(G)$ such that $G/X \in \mathcal{F}$ and $|X| \leq k$, it outputs an edge set Y of size at most $h(k) \cdot k$ and $G/Y \in \mathcal{F}$. Let $\alpha \geq 1$ be a real number. We now give an informal definition of α-approximate kernels. The kernelization algorithm takes an instance I with parameter k, runs in polynomial time, and produces a new instance I' with parameter k'. Both k' and the size of I' should be bounded in terms of just the parameter k. That is, there exists a function $g(k)$ such that $|I'| \leq g(k)$ and $k' \leq g(k)$. This function $g(k)$ is called the *size* of the kernel. For minimization problems, we also require the following from α-approximate kernels: For every $c \geq 1$, a c-approximate solution S' to I' can be transformed in polynomial time into a $(c \cdot \alpha)$-approximate solution S to I. However, if the quality of S' is "worse than" k', or $(c \cdot \alpha) \cdot OPT(I) > k$, the algorithm that transforms S' into S is allowed to fail. Here, $OPT(I)$ is the value of the optimum solution of the instance I.

Our first result is about CLIQUE CONTRACTION. It is known to be FPT. However, unless NP $\subseteq$ coNP/poly, it does not admit a polynomial kernel [8]. We show that it admits a PSAKS. That is, it admits a $(1 + \epsilon)$-approximate polynomial kernel with $O(k^{f(\epsilon)})$ vertices for every $\epsilon > 0$. In particular, we obtain the following result.

THEOREM 1.1. *For any $\epsilon > 0$, CLIQUE CONTRACTION parameterized by the size of solution k, admits a time efficient $(1 + \epsilon)$-approximate polynomial kernel with $O(k^{d+1})$ vertices, where $d = \lceil \frac{1}{\epsilon} \rceil$.*

Overview of the proof of Theorem 1.1. Let us fix an input (G, k) and a constant $\epsilon > 0$. Given a graph G, contracting edges of G to get into a graph class $\mathcal{F}$ is the same as partitioning the vertex set $V(G)$ into connected sets, $W_1, W_2, \ldots, W_\ell$, and then contracting each connected set to a vertex. These connected sets are called *witness sets*. A witness set W_i is called *non-trivial*, if $|W_i| \geq 2$, and *trivial* otherwise.

Observe that if a graph G can be transformed into a clique by contracting edges in F, then G can also be converted into a clique by deleting all the endpoints of edges in F. This observation implies that if G is k-contractible to a clique, then there exists an induced clique of size at least $|V(G)| - 2k$. Let I be a set of vertices in G, which induces this large clique and let $C = V(G) \setminus I$. Observe that C forms a vertex cover in the graph $\overline{G}$ (graph with vertex set $V(G)$ and those edges that are not present in $E(G)$). Using a factor 2-approximation algorithm, we find a vertex cover X of $\overline{G}$. Let $Y = V(\overline{G}) - X$ be an independent set in $\overline{G}$. If $|X| > 4k$, we immediately say No. Now, suppose that we have some solution and let $W_1, W_2, \ldots, W_\ell$ be those witness sets that are either non-trivial or contained in X. Now, let us say that a set W_i is *nice* if it has at least one vertex outside X, and *small*

if it contains less than $O(1/\epsilon)$ vertices. A set that is not *small* is *large*. Observe that there exists a $(1 + \epsilon)$-approximate solution where *the only sets* that are not nice are small. Also, observe that all nice sets are adjacent. Now, we classify all subsets of X of size at most $O(1/\epsilon)$ as *possible* and *impossible* small witness sets. Notice that if a set $A \subseteq X$ has more than $2k$ non-neighbors, then it can not possibly be a witness set, as one of these non-neighbors will be a trivial witness set. Now for every set, $A \subseteq X$ of size at most $O(1/\epsilon)$ mark all of its non-neighbors, but if there are more than $2k$, then mark $2k + 1$ of them. Now, look at an unmarked vertex in Y, the only reason it could still be relevant if it is part of some W_i. So its job is (*a*) connecting the vertices in W_i, or (*b*) potentially being the vertex in Y that is making some W_i nice, or (*c*) it is a neighbor to *all* the small (not nice) subsets of X in the solution. Now notice that any vertex in Y that is unmarked does jobs (*b*) and (*c*) equally well. So we only need to care about connectivity. Look at some nice and small set W_i; we only need to preserve the neighborhoods of the vertices of Y into W_i. For every subset of size $O(1/\epsilon)$, we keep one vertex in Y that has that set in its neighborhood. Notice that we do not care that different W_i's use different marked vertices for connectivity because merging two W_i's is more profitable for us. Finally, we delete all unmarked vertices and obtain an $(1 + \epsilon)$-approximate kernel of size roughly $k^{O(1/\epsilon)}$. We argue that this kernelization algorithm is time efficient i.e. the running time is polynomial in the size of an input and the constant in the exponent is independent of ϵ. This completes the overview of the proof for Theorem 1.1. Next, we move to Split Contraction.

Split Contraction is known to be W[1]-Hard [2]. We ask ourselves whether Split Contraction is *completely FPT-inapproximable* or admits an α-FPT-approximation algorithm, for some fixed constant $\alpha > 0$. We obtain two results towards our goal.

Theorem 1.2. *For every $\epsilon > 0$, Split Contraction admits a factor $(2 + \epsilon)$-FPT-approximation algorithm. In fact, for any $\epsilon > 0$, Split Contraction admits a $(2 + \epsilon)$-approximate kernel with $O(k^{f(\epsilon)})$ vertices.*

Given, Theorem 1.2, it is natural to ask whether Split Contraction admits a factor $(1 + \epsilon)$-FPT-approximation algorithm, for every $\epsilon > 0$. We show that this is not true and obtain the following hardness result.

Theorem 1.3. *Assuming Gap-ETH, no FPT time algorithm can approximate Split Contraction within a factor of $\left(\frac{5}{4} - \delta\right)$, for any fixed constant $\delta > 0$.*

Overview of the proofs of Theorems 1.2 and 1.3. Our proof for Theorem 1.2 uses ideas for $(1 + \epsilon)$-approximate kernel for Clique Contraction (Theorem 1.1) and thus we omit its overview. Towards the proof of Theorem 1.3, we give a gap preserving reduction from a variant of the Densest-k-Subgraph problem (given a graph G and an integer k, find a subset $S \subseteq V(G)$ of k vertices that induces maximum number of edges). Chalermsook et al. [13] showed that, assuming Gap-ETH[1], for any $g = o(1)$, there is no FPT-time algorithm that, given an integer k and any graph G on n vertices that contains at least one k-clique, always output $S \subseteq V(G)$, of size k, such that $\text{Den}(S) \geq k^{-g(k)}$. Here, $\text{Den}(S) = |E(G[S])|/\binom{|S|}{2}$. We need a strengthening of this result that says that assuming Gap-ETH, for any $g = o(1)$ and for any constant $\alpha > 1$, there is no FPT-time algorithm that, given an integer k and any graph G on n vertices that contains at least one k-clique, always outputs $S \subseteq V(G)$, of size αk, such that $\text{Den}(S) \geq k^{-g(k)}$. Starting from this result, we give a gap-preserving reduction to Split Contraction that takes FPT time and obtain Theorem 1.3. Given an instance (G, k) of Densest-k-Subgraph, we first use color coding to partition the edges into $t = \binom{k}{2}$ color classes such that every color class contains exactly one edge of a "densest subgraph" (or a clique). For each color class we make one *edge selection gadget*. Each edge selection gadget corresponding

---

[1]We refer the readers to [13] for the definition of Gap-ETH and related terms.

to the color class j consists of an independent set ES_j that contains a vertex corresponding to each edge in the color class j, and a *cap vertex* g_j that is adjacent to every vertex in ES_j. Next, we add a sufficiently large clique Z of size $\rho \cdot |V(G)|$, where for every vertex $v \in V(G)$, we have ρ vertices. Every vertex in an edge selection gadget is adjacent to every vertex of Z, except those corresponding to the endpoints of the edge the vertex represents. Finally, we add a clique SV of size t that has one vertex s_j for each edge selection gadget. Make the vertex s_j adjacent to every vertex in ES_j. We also add sufficient guards on vertices everywhere, so that "unwanted" contractions do not happen. The idea of the reduction is to contract edges in a way that the vertices in SV, Z, and g_j, $j \in \{1, \ldots, t\}$, become a giant clique and other vertices become part of an independent set, resulting in a split graph. Towards this we first use $2t$ contractions so that g_j, s_j, and a vertex $a_j \in \mathsf{ES}_j$ are contracted into one. One way to ensure that they form a clique along with Z is to contract each of them to a vertex in Z. However, this will again require t edge contractions. We set our budget in a way that this is not possible. Thus, what we need is to destroy the non-neighbors of a_j. One way to do this again will be to match the vertices obtained after the first round of $2t$ contractions in a way that there are no non-adjacencies left. However, this will also cost $t/2$, and our budget does not allow this. The other option (which we take) is to take the union of all non-neighbors of a_j, say N, and contract each of them to one of the vertex in $Z \setminus N$. Observe that to minimize the contractions to get rid of non-neighbors of a_j, we would like to minimize $|N|$. This will happen when N spans a large number of edges. Thus, it precisely captures the Densest-k-Subgraph problem. The budget is chosen in a way that we get the desired gap-preserving reduction, which enables us to prove Theorem 1.3.

Our final result concerns Chordal Contraction. Lokshtanov et al. [38] showed that Chordal Contraction is W[2]-Hard. We observe that the existing W[2]-hardness reduction can be adapted to show the following theorem.

Theorem 1.4. *Assuming FPT$\neq$ W[1], no FPT time algorithm can approximate* Chordal Contraction *within a factor of $F(k)$. Here, $F(k)$ is a function depending on k alone.*

Overview of the proof of Theorem 1.4. Towards proving Theorem 1.4, we give a 1-approximate polynomial parameter transformation (1-appt) from Set Cover (given a universe U, a family of subsets $\mathcal{S}$, and an integer k, we shall decide the existence of a subfamily of size k that contains all the elements of U) to Chordal Contraction. That is, given any solution of size at most ℓ for Chordal Contraction, we can transform this into a solution for Set Cover of size at most ℓ. Karthik et al. [35] showed that assuming FPT$\neq$ W[1], no FPT time algorithm can approximate Set Cover within a factor of $F(k)$. Pipelining this result with our reduction we get Theorem 1.4.

Related Work. To the best of our knowledge, Heggernes et al. [33] was the first to explicitly study $\mathcal{F}$-Contraction from the viewpoint of Parameterized Complexity. They showed that in case $\mathcal{F}$ is the family of trees, $\mathcal{F}$-Contraction is FPT but does not admit a polynomial kernel, while in case $\mathcal{F}$ is the family of paths, the corresponding problem admits a faster algorithm and an $O(k)$-vertex kernel. Golovach et al. [29] proved that if $\mathcal{F}$ is the family of planar graphs, then $\mathcal{F}$-Contraction is again FPT. Moreover, Cai and Guo [8] showed that in case $\mathcal{F}$ is the family of cliques, $\mathcal{F}$-Contraction is solvable in time $2^{O(k \log k)} \cdot n^{O(1)}$, while in case $\mathcal{F}$ is the family of chordal graphs, the problem is W[2]-Hard. Heggernes et al. [34] developed an FPT algorithm for the case where $\mathcal{F}$ is the family of bipartite graphs. Later, a faster algorithm was proposed by Guillemot and Marx [31].

Pioneering work of Lokshtanov et al. [40] on the approximate kernel is being followed by a series of papers generalizing/improving results mentioned in this work and establishing lossy kernels for various other problems. Lossy kernels for some variations of Connected Vertex Cover [22, 36],

Connected Feedback Vertex Set [44], Steiner Tree [21] and Dominating Set [23, 45] have been established (also see [41, 46]). Krithika et al. [37] were the first to study graph contraction problems from the lenses of lossy kernelization. They proved that for any $\alpha > 1$, Tree Contraction admits an α-lossy kernel with $O(k^d)$ vertices, where $d = \lceil \alpha/(\alpha - 1) \rceil$. Agarwal et al. [1] proved similar result for $\mathcal{F}$-Contraction problems where graph class $\mathcal{F}$ is defined in parametric way from set of trees. Eiben et al. [22] obtained similar result for Connected $\mathcal{H}$-Hitting Set problem.

Guide to the paper. We start by giving the notations and preliminaries that we use throughout the paper in Section 2. This section is best used as a reference, rather than being read linearly. In Section 3 we give the $(1 + \epsilon)$–approximate polynomial kernel for Clique Contraction. Section 4 gives the $(2 + \epsilon)$–approximate polynomial kernel for Split Contraction. The ideas here are similar to those used in Section 3, and thus an eager reader could skip further. In Section 5, we show that assuming Gap-ETH, no FPT time algorithm can approximate Split Contraction within a factor of $\left(\frac{5}{4} - \delta\right)$, for any fixed constant $\delta > 0$. Section 6 shows that, assuming FPT$\neq$ W[1], no FPT time algorithm can approximate Chordal Contraction within a factor of $F(k)$. This is an adaptation of the existing W[2]-hardness reduction and may be skipped. Thus, our main technical results appear in Sections 3 and 5. We conclude the paper with some interesting open problem in Section 7.

2 PRELIMINARIES

In this section, we give notations and definitions that we use throughout the paper. Unless specified, we will be using all general graph terminologies from the book of Diestel [16].

2.1 Graph Theoretic Definitions and Notations

For an undirected graph G, sets $V(G)$ and $E(G)$ denote the set of vertices and edges, respectively. Two vertices u, v in $V(G)$ are said to be *adjacent* if there is an edge uv in $E(G)$. The neighborhood of a vertex v, denoted by $N_G(v)$, is the set of vertices adjacent to v in G. For subset S of vertices, we define $N(S) = \bigcup_{v \in S} N(v)) \setminus S$. The subscript in the notation for the neighborhood is omitted if the graph under consideration is clear. For a set of edges F, set $V(F)$ denotes the endpoints of edges in F. For a subset S of $V(G)$, we denote the graph obtained by deleting S from G by $G - S$ and the subgraph of G induced on set S by $G[S]$. For two subsets S_1, S_2 of $V(G)$, we say S_1, S_2 *are adjacent if there exists an edge* with one endpoint in S_1 and other in S_2. A set S is connected set if any two vertices in it are connected by a path which is completely in S. We say a graph is *connected* if $V(G)$ is a connected set.

An edge e in G is a *chord* of a cycle C (resp. path P) if (i) both the endpoints of e are in C (resp. in P), and (ii) edge e is not in C (resp. not in P). An *induced cycle* (resp. *path*) is a cycle (resp. path) which has no chord. We denote induced cycle and path on ℓ vertices by C_ℓ and P_ℓ, respectively. A *complete graph G* is an undirected graph in which for every pair of vertices $u, v \in V(G)$, there is an edge uv in $E(G)$. As an immediate consequence of definition we get the following.

Lemma 2.1. *A connected graph G is complete if and only if G does not contain an induced P_3.*

A *clique* is a subset of vertices in the graph that induces a complete graph. A set $I \subseteq V(G)$ of pairwise non-adjacent vertices is called an *independent set*. A graph G is a *split graph* if $V(G)$ can be partitioned into a clique and an independent set. For split graph G, partition (X, Y) is *split partition* if X is a clique and Y is an independent set. In this article, whenever we mention a split partition, we first mention the clique followed by the independent set. We will also use the following well-known characterization of split graphs. Let, $2K_2$ be a graph induced on four vertices, which contains exactly two edges and no isolated vertices.

LEMMA 2.2 ([30]). *A graph G is a split graph if and only if it does not contain C_4, C_5 or $2K_2$ as an induced subgraph.*

A graph G is *chordal* if every induced cycle in G is a triangle; equivalently, if every cycle of length at least four has a chord. A vertex subset $S \subseteq V(G)$ is said to *cover* an edge $uv \in E(G)$ if $S \cap \{u, v\} \neq \emptyset$. A vertex subset $S \subseteq V(G)$ is called a *vertex cover* in G if it covers all the edges in G.

We start with the following observation, which is useful to find a large induced clique in the input graph. The *complement* of G, denoted by $\bar{G}$, is a graph whose vertex set is $V(G)$ and edge set is precisely those edges which are not present in $E(G)$. Note that given a graph G, if S is a set of vertices such that $G - S$ is a clique, then S is a vertex cover in the complement graphs of G, denoted by $\bar{G}$, as $\bar{G} - S$ is edgeless. Using the well-known factor 2-approximation algorithm for VERTEX COVER [4], we have following.

OBSERVATION 2.1 ([4]). *There is a factor 2-approximation algorithm to compute a set of vertices whose deletion results in a complete graph.*

Using, Lemma 2.2 one can obtain a simple factor 5-approximation algorithm for deleting vertices to get a split graph.

OBSERVATION 2.2. *There is a factor 5-approximation algorithm to compute a set of vertices whose deletion results in a split graph.*

Recently, for every $\epsilon > 0$, a factor $(2 + \epsilon)$-approximation algorithm for deleting vertices to get a split graph has been obtained [39]. However, for our purposes Observation 2.2 will suffice.

2.2 Graph Contraction

The *contraction* of edge $e = uv$ in G deletes vertices u and v from G, and adds a new vertex, which is made adjacent to vertices that were adjacent to either u or v. Any parallel edges added in the process are deleted so that the graph remains simple. The resulting graph is denoted by G/e. Formally, for a given graph G and edge $e = uv$, we define G/e in the following way: $V(G/e) = (V(G) \cup \{w\}) \setminus \{u, v\}$ and $E(G/e) = \{xy \mid x, y \in V(G) \setminus \{u, v\}, xy \in E(G)\} \cup \{wx \mid x \in N_G(u) \cup N_G(v)\}$. For a subset of edges F in G, graph G/F denotes the graph obtained from G by contracting each connected component of the subgraph $G' = (V(F), F)$ to a single vertex. Formally, $V(G/F) = (V(G) \cup w) \setminus V(F)$ and $E(G/F) = \{xy \mid x, y \in V(G) \setminus V(F), xy \in E(G)\} \cup \{wx \mid x \in N_G(V(F))\}$. We say that a graph G is *contractible* to a graph H if there exists an onto function $\psi : V(G) \rightarrow V(H)$ such that the following properties hold.

- For any vertex h in $V(H)$, graph $G[W(h)]$ is connected, where set $W(h) := \{v \in V(G) \mid \psi(v) = h\}$.
- For any two vertices h, h' in $V(H)$, edge hh' is present in H if and only if there exists an edge in G with one endpoint in $W(h)$ and another in $W(h')$.

For a vertex h in H, set $W(h)$ is called a *witness set* associated with h. We define H-*witness structure* of G, denoted by $\mathcal{W}$, as collection of all witness sets. Formally, $\mathcal{W} = \{W(h) \mid h \in V(H)\}$. Witness structure $\mathcal{W}$ is a partition of vertices in G, where each witness forms a connected set in G. Recall that if a *witness set* contains more than one vertex, then we call it *non-trivial* witness set, otherwise a *trivial* witness set.

If graph G has a H-witness structure, then graph H can be obtained from G by a series of edge contractions. For a fixed H-witness structure, let F be the union of spanning trees of all witness sets. By convention, the spanning tree of a singleton set is an empty set. Thus, to obtain H from G, it is *sufficient* to contract edges in F. If such witness structure exists, then we say that graph G is contractible to H. Note that, for a contraction set of edges F, the witness structure can be obtained

uniquely by the collection of maximal connected sets in $G' = (V(F), F)$. We say that graph G is
k-*contractible* to H if cardinality of F is at most k. In other words, H can be obtained from G by at
most k edge contractions. Following observation is an immediate consequence of definitions.

OBSERVATION 2.3. *If graph G is k-contractible to graph H, then the following statements are true.*
- *For any witness set W in a H-witness structure of G, the cardinality of W is at most $k + 1$.*
- *For a fixed H-witness structure, the number of vertices in G, which are contained in non-trivial
  witness sets is at most $2k$.*

In the following two observations, we state that if a graph can be transformed into a clique or a
split graph by contracting few edges, then it can also be converted into a clique or split graph by
deleting few vertices.

OBSERVATION 2.4. *If a graph G is k-contractible to a clique, then G can be converted into a clique
by deleting at most $2k$ vertices.*

PROOF. Let F be a set of edges of size at most k such that G/F is a clique. Let $\mathcal{W}$ be a G/F-witness
structure of G. Let X be a set of all vertices which are contained in the non-trivial witness sets in
$\mathcal{W}$. By Observation 2.3, size of X is at most $2k$. Any two vertices in $V(G) \setminus X$ are adjacent to each
other as these vertices form singleton sets, which are adjacent in G/F. Hence, G can be converted
into a clique by deleting vertices in X.   □

OBSERVATION 2.5. *If a graph G is k-contractible to a split graph then G can be converted into a
split graph by deleting at most $2k$ vertices.*

PROOF. For graph G, let F be the set of edges such that G/F is a split graph and $|F| \le k$. Let $V(F)$
be the collection of all endpoints of edges in F. Since cardinality of F is at most k, $|V(F)|$ is at most
$2k$. We argue that $G - V(F)$ is a split graph. For the sake of contradiction, assume that $G - V(F)$ is
not a split graph. We know that a graph is split if and only if it does not contain induced C_4, C_5
or $2K_2$. This implies that there exists a set of vertices V' in $V(G) \setminus V(F)$ such that $G[V']$ is either
C_4, C_5 or $2K_2$. Since no edge in F is incident on any vertices in V', $G/F[V']$ is isomorphic to $G[V']$.
Hence, there exists a C_4, C_5 or $2K_2$ in G/F contradicting the fact that G/F is a split graph. Hence,
our assumption is wrong and $G - V(F)$ is a split graph.   □

Consider a connected graph G which is k-contractible to the clique K_ℓ. Let $\mathcal{W}$ be a K_ℓ-witness
structure of G. The following observation gives a sufficient condition for obtaining a witness
structure of an induced subgraph of G from $\mathcal{W}$.

OBSERVATION 2.6. *Let $\mathcal{W}$ be a clique witness structure of G. If there exists two different witness
sets $W(t_1), W(t_2)$ in $\mathcal{W}$ and a vertex v in $W(t_1)$ such that the set $W(t) = (W(t_1) \cup W(t_2)) \setminus \{v\}$ is a
connected set in $G - \{v\}$, then $\mathcal{W}'$ is a clique witness structure of $G - \{v\}$, where $\mathcal{W}'$ is obtained from
$\mathcal{W}$ by removing $W(t_1), W(t_2)$ and adding $W(t)$.*

PROOF. Let $G' = G - \{v\}$. Note that $\mathcal{W}'$ is a partition of vertices in G'. Any set in $\mathcal{W}' \setminus \{W(t)\}$
is a witness set in $\mathcal{W}$ and does not contain v. Hence, these sets are connected in G'. Since $G'[W(t)]$
is also connected, all the witness sets in $\mathcal{W}'$ are connected in G'.
Consider any two witness sets $W(t'), W(t'')$ in $\mathcal{W}'$. If none of these two is equal to $W(t)$ then
both of these sets are present in $\mathcal{W}$. Since none of these witness sets contains vertex v, they are
adjacent to each other in G'. Now, consider a case when one of them, say $W(t'')$, is equal to $W(t)$.
As witness sets $W(t')$ and $W(t_2)$ are present in $\mathcal{W}$, there exists an edge with one endpoint in $W(t')$
and another in $W(t_2)$. The same edge is present in graph G' as it is not incident on v. Since $W(t_2)$
is subset of $W(t)$, sets $W(t')$ and $W(t)$ are adjacent in G'. Hence any two witness sets in $\mathcal{W}'$ are
adjacent to each other. This implies that $\mathcal{W}'$ is a clique witness structure of graph $G - \{v\}$.   □

In the case of SPLIT CONTRACTION, the following observation guarantees the existence of witness structure with a particular property.

OBSERVATION 2.7. *For a connected graph G, let F be a set of edges such that G/F is a split graph. Then, there exists a set of edges F' which satisfies the following properties: (i) G/F' is a split graph. (ii) The number of edges in F' is at most $|F|$. (iii) There exists a split partition of G/F' such that all vertices in G/F' which correspond to a non-trivial witness set in G/F'-witness structure of G are in clique side.*

PROOF. Let (C, I) be a split partition of vertices of G/F such that C is a clique and I is an independent set. If all the vertices corresponding to non-trivial witness sets are in C, then the observation is true. Consider a vertex a in I which corresponds to a non-trivial witness set W_a. Since G is connected, G/F is a connected split graph. This implies that there exists a vertex, say b, in C which is adjacent to a in G/F. We denote witness set corresponding to b by W_b. We construct a new witness structure by shifting all but one vertices in W_a to W_b. Since ab is an edge in G/F, there exists an edge in G with one endpoint in W_a and another in W_b. Let that edge be $u_a u_b$ with vertices u_a and u_b contained in sets W_a and W_b, respectively. Consider a spanning tree T of graph $G[W_a]$ which is rooted at u_a. We can replace edges in F whose both endpoints are in $V(W_a)$ with $E(T)$ to obtain another set of edges F^* such that G/F^* is a split graph. Formally, $F^* = (F \cup E(T)) \setminus (E(G[W_a]) \cap F)$. Note that the number of edges in F^* and F are same. Let v_1 be a leaf vertex in T and v_2 be its unique neighbor. Consider $F' = (F^* \cup \{u_a u_b\}) \setminus \{v_1 v_2\}$. Since edge $v_1 v_2$ is in F^* and $u_a u_b$ is not in F^*, $|F'| = |F^*|$. We now argue that G/F' is also a split graph. Let $\mathcal{W}'$ be the G/F'-witness structure of G. Note that $\mathcal{W}'$ can be obtained from G/F^*-witness structure $\mathcal{W}^*$ of G by replacing W_a by $\{v_1\}$ and W_b by $W_b \cup (W_a \setminus \{v_1\})$. Since all other witness sets remains unchanged any witness set which was adjacent to W_b is also adjacent to $W_b \cup (W_a \setminus \{v_1\})$. Similarly, any witness set which was not adjacent to W_a is not adjacent to $\{v_2\}$. In other words, this operation of shifting edges did not remove any vertex from the neighborhood of b (which is in C) nor it added any vertex in the neighborhood of a (which is in I). Hence, G/F' is also a split graph with (C, I) as one of its split partition. Note that there exists a split partition of G/F' such that the number of vertices in the independent side corresponding to non-trivial witness set is one less than the number of vertices in I which corresponds to non-trivial witness sets. Hence, by repeating this process at most $|V(G)|$ times, we get a set of edges that satisfy three properties mentioned in the observation. □

2.3 Parameterized Complexity and Lossy Kernelization

An important notion in parameterized complexity is *kernelization*, which captures the efficiency of data reduction techniques. A parameterized problem Π admits a kernel of size $g(k)$ (or $g(k)$-kernel) if there is a polynomial time algorithm (called *kernelization algorithm*) which takes as input (I, k), and returns an instance (I', k') of Π such that: (i) (I, k) is a yes-instance if and only if (I', k') is a yes-instance; and (ii) $|I'| + k' \leq g(k)$, where $g(\cdot)$ is a computable function whose value depends only on k. Depending on whether the function $g(\cdot)$ is *linear, polynomial* or *exponential*, the problem is said to admit a *linear, polynomial* or *exponential kernel*, respectively. We refer to the corresponding chapters in the books [15, 17, 24, 26, 43] for a detailed introduction to the field of kernelization.

In lossy kernelization, we work with the optimization analog of parameterized problem. Along with an instance and a parameter, an optimization analog of the problem also has a string called *solution*. We start with the definition of a *parameterized optimization problem*. It is the parameterized analog of an optimization problem used in the theory of approximation algorithms.

DEFINITION 2.1 (PARAMETERIZED OPTIMIZATION PROBLEM). *A parameterized optimization problem is a computable function $\Pi : \Sigma^* \times \mathbb{N} \times \Sigma^* \mapsto \mathbb{R} \cup \{\pm\infty\}$. The instances of Π are pairs $(I, k) \in \Sigma^* \times \mathbb{N}$ and a solution to (I, k) which is simply a string $S \in \Sigma^*$ such that $|S| \leq |I| + k$.*

The *value* of a solution S is $\Pi(I, k, S)$. In this paper, all optimization problems are minimization problems. Therefore, we present the rest of the section only with respect to parameterized *minimization* problem. The *optimum value* of (I, k) is defined as:

$$\text{OPT}_\Pi(I, k) = \min_{S \in \Sigma^*, \, |S| \le |I| + k} \Pi(I, k, S),$$

and an *optimum solution* for (I, k) is a solution S such that $\Pi(I, k, S) = \text{OPT}_\Pi(I, k)$. For a constant $c > 1$, S is *c-factor approximate* solution for (I, k) if $\frac{\Pi(I,k,S)}{OPT_\Pi(I,k)} \le c$. We omit the subscript Π in the notation for optimum value if the problem under consideration is clear from the context.

For some parameterized optimization problems we are unable to obtain FPT algorithms, and we are also unable to find satisfactory polynomial time approximation algorithms. In this case one might aim for FPT-approximation algorithms, algorithms that run in time $f(k)n^c$ and provide good approximate solutions to the instance.

DEFINITION 2.2. *Let $\alpha \ge 1$ be constant. A fixed parameter tractable α-approximation algorithm for a parameterized optimization problem Π is an algorithm that takes as input an instance (I, k), runs in time $f(k)|I|^{O(1)}$, and outputs a solution S such that $\Pi(I, k, S) \le \alpha \cdot \text{OPT}(I, k)$ if Π is a minimization problem, and $\alpha \cdot \Pi(I, k, S) \ge \text{OPT}(I, k)$ if Π is a maximization problem.*

Note that Definition 2.2 only defines constant factor FPT-approximation algorithms. The definition can in a natural way be extended to approximation algorithms whose approximation ratio depends on the parameter k, on the instance I, or on both. Next, we define an *α-approximate polynomial-time preprocessing algorithm* for a parameterized minimization problem Π as follows.

DEFINITION 2.3 (α-APPROXIMATE POLYNOMIAL-TIME PREPROCESSING ALGORITHM). *Let $\alpha \ge 1$ be a real number and Π be a parameterized minimization problem. An α-approximate polynomial-time preprocessing algorithm is defined as a pair of polynomial-time algorithms, called the* reduction algorithm *and the* solution lifting algorithm, *that satisfy the following properties.*

- *Given an instance (I, k) of Π, the reduction algorithm computes an instance (I', k') of Π.*
- *Given instances (I, k) and (I', k') of Π, and a solution S' to (I', k'), the solution lifting algorithm computes a solution S to (I, k) such that $\frac{\Pi(I,k,S)}{OPT(I,k)} \le \alpha \cdot \frac{\Pi(I',k',S')}{OPT(I',k')}$.*

We sometimes refer α-approximate polynomial-time preprocessing algorithm kernel as *α-lossy rule* or *α-reduction rule*.

DEFINITION 2.4 (PSAKS). *Let Π be a parameterized optimization problem. A polynomial size approximate kernelization scheme(PSAKS) for Π is a family of α-approximate polynomial kernelization algorithms, with one such algorithm for every $\alpha > 1$.*

3 LOSSY KERNEL FOR CLIQUE CONTRACTION

In this section, we present a lossy kernel for CLIQUE CONTRACTION. We first define a natural optimization version of the problem.

$$\text{CLC}(G, k, F) = \begin{cases} \min\{|F|, k + 1\} & \text{if } G/F \text{ is a clique} \\ \infty & \text{otherwise} \end{cases}$$

If the number of vertices in the input graph is at most $k + 3$, then we can return the same instance as a kernel for the given problem. Further, we assume that the input graph is connected; otherwise, it can not be edited into a clique by edge contraction only. Thus, we only consider *connected graphs with at least $k + 3$ vertices*. By the definition of optimization problem, for any set of edges F, if G/F is a clique, then the maximum value of $\text{CLC}(G, k, F)$ is $k + 1$. Hence, any spanning tree of G is a solution of cost $k + 1$. We call it a *trivial solution* for the given instance. Consider an instance $(P_4, 1)$,

where P_4 is a path on four vertices. One needs to contract at least two edges to convert P_4 into a clique. We call $(P_4, 1)$ a *trivial No-instance* for this problem. Finally, we assume that we are given an $\epsilon > 0$.

We start with a reduction rule, which says that if the minimum number of vertices that need to be deleted from an input graph to obtain a clique is large, then we can return a trivial instance as a lossy kernel.

REDUCTION RULE 3.1. *For a given instance (G, k), apply the algorithm mentioned in Observation 2.1 to find a set X such that $G - X$ is a clique. If the size of X is greater than $4k$, then return $(P_4, 1)$.*

LEMMA 3.1. *Reduction Rule 3.1 is a 1-reduction rule.*

PROOF. Let (G, k) be an instance of CLIQUE CONTRACTION such that the Reduction Rule 3.1 returns $(P_4, 1)$ when applied on it. The solution lifting algorithm returns a spanning tree F of G. Note that for a set of edges F', if P_4/F' is a clique then F' contains at least two edges. This implies $\text{CLC}(P_4, 1, F') = 2$ and $\text{OPT}(P_4, 1) = 2$.

Since a factor 2-approximation algorithm returned a set of size strictly more than $4k$, for any set X' of size at most $2k$, $G - X'$ is not a clique. But by Observation 2.4, if G is k-contractible to a clique then G can be edited into a clique by deleting at most $2k$ vertices. Hence, for any set of edges F^* if G/F^* is a clique, then the size of F^* is at least $k + 1$. This implies that $\text{OPT}(G, k) = k + 1$, and for a spanning tree F of G, $\text{CLC}(G, k, F) = k + 1$.

Combining these values, we get $\frac{\text{CLC}(G,k,F)}{\text{OPT}(G,k)} = \frac{k+1}{k+1} = \frac{2}{2} = \frac{\text{CLC}(P_4,1,F')}{\text{OPT}(P_4,1)}$. This implies that if F' is factor c-approximate solution for $(P_4, 1)$, then F is factor c-approximate solution for (G, k). This concludes the proof. □

We now consider an instance (G, k) for which Reduction Rule 3.1 does not return a trivial instance. This implies that for a given graph G, in polynomial time, one can find a partition (X, Y) of $V(G)$ such that $G - X = G[Y]$ is a clique and $|X|$ is at most $4k$. For $\epsilon > 0$, find a smallest integer d, such that $\frac{d+1}{d} \leq 1 + \epsilon$. In other words, fix $d = \lceil \frac{1}{\epsilon} \rceil$. We note that if the number of vertices in the graph is at most $O(k^{d+1})$, then the algorithm returns this graph as a lossy kernel of the desired size. Hence, without loss of generality, we assume that the number of vertices in the graph is larger than $O(k^{d+1})$.

Given an instance (G, k), a partition (X, Y) of $V(G)$ with $G[Y]$ being a clique, and an integer d, consider the following two marking schemes.

MARKING SCHEME 3.1. *For a subset A of X, let $M_1(A)$ be the set of vertices in Y whose neighborhood contains A. For every subset A of X which is of size at most d, mark a vertex in $M_1(A)$.*

Formally, $M_1(A) = \{y \in Y | A \subseteq N(y)\}$. If $M_1(A)$ is an empty set, then the marking scheme does not mark any vertex. If it is non-empty, then the marking scheme arbitrarily chooses a vertex and marks it.

MARKING SCHEME 3.2. *For a subset A of X, let $M_2(A)$ be the set of vertices in Y whose neighborhood does not intersect A. For every subset A of X which is of size at most d, mark $2k + 1$ vertices in $M_2(A)$.*

Formally, $M_2(A) = \{y \in Y | N(y) \cap A = \emptyset\}$. If the number of vertices in $M_2(A)$ is at most $2k + 1$, then the marking scheme marks all vertices in $M_2(A)$. If it is larger than $2k + 1$, then it arbitrarily chooses $2k + 1$ vertices and marks them.

REDUCTION RULE 3.2. *For a given instance (G, k), partition (X, Y) of $V(G)$ with $G[Y]$ being a clique, and an integer d, apply the Marking Schemes 3.1 and 3.2. Let G' be the graph obtained from G by deleting all the unmarked vertices in Y. Return the instance (G', k).*
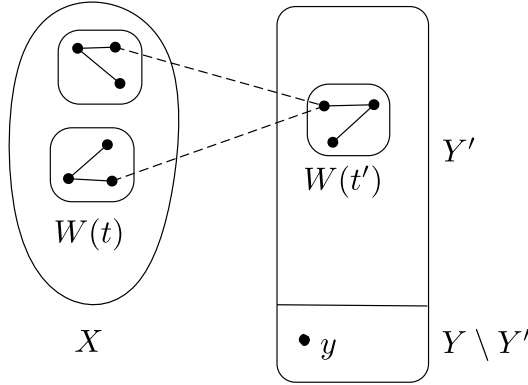
Fig. 1. Straight lines (e.g within $W(t)$) represent edges in original solution F. Dashed lines (e.g. across $W(t)$ and $W(t')$) represents extra edges added to solution F. Please refer to the proof of Lemma 3.3.

Above reduction rule can be applied in time $|X|^d \cdot |V(G)|^{O(1)} = O(k^{O(d)}|V(G)|^{O(1)})$ as $|X|$ is at most $4k$. Note that G' is an induced subgraph of G. We first show that since G is a connected graph, G' is also connected. In the following lemma, we prove a stronger statement.

LEMMA 3.2. *Consider instance* (G, k) *of* CLIQUE CONTRACTION. *Let* Y' *be the set of vertices marked by Marking Scheme 3.1 or 3.2 for some positive integer d. For any subset* Y'' *of* $Y \setminus Y'$, *let* G'' *be the graph obtained from* G *by deleting* Y''. *Then,* G'' *is connected.*

PROOF. Recall that, by our assumption, G is connected and Y induces a clique in G. Hence, for every vertex in X, there exists a path from it to some vertex in Y. By the construction of G'', $(X, Y \setminus Y'')$ forms a partition of $V(G'')$ and $Y \setminus Y''$ is a clique in G''. To prove that G'' is connected, it is sufficient to prove that for every vertex in X, there exists a path from it to a vertex in $Y \setminus Y''$ in G.

Fix an arbitrary vertex, say x, in X. Consider a path P from x to y in G, where y is some vertex in Y. Without loss of generality, we can assume that y is the only vertex in $V(P) \cap Y$. We argue that there exists another path, say P_1, from x to a vertex in $Y \setminus Y''$. If y is in $Y \setminus Y''$ then $P_1 = P$ is a desired path. Consider the case when y is in Y''. Let x_0 be the vertex in $V(P)$ which is adjacent with y. Note that x_0 may be the same as x. As Marking Scheme 3.1 considers all subsets of size at most d, it considered singleton set $\{x_0\}$. As x_0 is adjacent with y, we have $\{x_0\} \subseteq N(y)$. Since y is in Y'', and hence unmarked, there exists a vertex, say y_1, in Y which has been marked by Marking Scheme 3.1. Consider a path P_1 obtained from P by deleting vertex y (and hence edge x_0y) and adding vertex y_1 with edge x_0y_1. This is a desired path from x to a vertex in $Y \setminus Y''$. As x is an arbitrary vertex in X, this statement is true for any vertex in X and hence G'' is connected.   □

Thus, because of Lemma 3.2, from now onwards, we assume that G' is connected. In fact, in one of our proofs, we will iteratively remove vertices from $Y \setminus Y'$, and Lemma 3.2 ensures that the graph at each step remains connected. In the following lemma, we argue that given a solution for (G', k), we can construct a solution of almost the same size for (G, k).

LEMMA 3.3. *Let* (G', k) *be the instance returned by Reduction Rule 3.2 when applied on an instance* (G, k). *If there exists a set of edges of size at most* k, *say* F', *such that* G'/F' *is a clique, then there exists a set of edges* F *such that* G/F *is a clique and cardinality of* F *is at most* $(1 + \epsilon) \cdot |F'|$.

PROOF. If no vertex in Y is deleted, then G' and G are identical graphs, and the statement is true. We assume that at least one vertex in Y is deleted. Let Y' be the set of vertices in Y, which are marked. Note that the sets X, Y' forms a partition of $V(G')$ such that Y' is a clique and a proper subset of Y. Let $\mathcal{W}'$ be a G'/F'-witness structure of G'. We construct a clique witness structure $\mathcal{W}$ of G from $\mathcal{W}'$ by adding singleton witness sets $\{y\}$ for every vertex y in $Y \setminus Y'$. Since $G[Y \setminus Y']$ is a clique in G, any two newly added witness sets are adjacent to each other. Moreover, any witness set in $\mathcal{W}'$, which intersects Y' is also adjacent to the newly added witness sets. We now consider witness sets in $\mathcal{W}'$, which do not intersect Y'.

Let $\mathcal{W}^\star$ be a collection of witness sets $W(t)$ in $\mathcal{W}'$ such that $W(t)$ is contained in X and there exists a vertex y in $Y \setminus Y'$ whose neighborhood does not intersect with $W(t)$. See Figure 1. We argue that every witness set in $\mathcal{W}^\star$ has at least $d + 1$ vertices. For the sake of contradiction, assume that there exists a witness set $W(t)$ in $\mathcal{W}^\star$ which contains at most d vertices. Since Marking Scheme 3.2 iterated over all the subsets of X of size at most d, it also considered $W(t)$ while marking. Note that the vertex y belongs to the set $M_2(W(t))$. Since y is unmarked, there are $2k + 1$ vertices in $M_2(W(t))$ which have been marked. All these marked vertices are in G'. Since the cardinality of F' is at most k, the number of vertices in $V(F')$ is at most $2k$. Hence, at least one marked vertex in $M_2(W(t))$ is a singleton witness set in $\mathcal{W}'$. However, there is no edge between this singleton witness set and $W(t)$. This non-existence of an edge contradicts the fact that any two witness sets in $\mathcal{W}'$ are adjacent to each other in G'. Hence, our assumption is wrong, and $W(t)$ has at least $d + 1$ vertices.

Next, we show that there exists a witness set in $\mathcal{W}'$ that intersects Y'. This is ensured by the fact that G' is connected, and we are in the case where at least one vertex in Y is deleted. The last assertion implies that Y' is non-empty, and hence there must be a witness set in $\mathcal{W}'$ that intersects Y'. Let $W(t')$ be a witness set in $\mathcal{W}'$ that intersects Y'. Note that $W(t')$ is adjacent to every vertex in $Y \setminus Y'$. Let $W(t)$ be a witness set in $\mathcal{W}^\star$. Since $W(t')$ and $W(t)$ are two witness sets in the G'/F'-witness structure, there exists an edge with one endpoint in $W(t')$ and another in $W(t)$. Therefore, the set $W(t') \cup W(t)$ is adjacent to every other witness set in $\mathcal{W}$.

We now describe how to obtain F from F'. We initialize $F = F'$. For every witness set $W(t)$ in $\mathcal{W}^\star$ add an edge between $W(t)$ and $W(t')$ to the set F'. Equivalently, we construct a new witness set by taking the union of $W(t')$ and all witness sets $W(t)$ in $\mathcal{W}^\star$. This witness set is adjacent to every vertex in $Y \setminus Y'$, and hence G/F is a clique. We now argue the size bound on F. Note that we have added one extra edge for every witness set $W(t)$ in $\mathcal{W}^\star$. We also know that every such witness set has at least $d + 1$ vertices. Hence, we have added one extra edge for at least d edges in the solution F'. Moreover, since witness sets in $\mathcal{W}^\star$ are vertex disjoint, no edge in F can be part of two witness sets. This implies that the number of edges in F' is at most $(d+1/d)|F| \leq (1 + \epsilon) \cdot |F|$. □

In the following lemma, we argue that the value of the optimum solution for the reduced instance can be upper bounded by the value of an optimum solution for the original instance.

LEMMA 3.4. *Let (G', k) be the instance returned by Reduction Rule 3.2 when applied on an instance (G, k). If $OPT(G, k) \leq k$, then $OPT(G', k) \leq OPT(G, k)$.*

PROOF. Let F be a set of at most k edges in G such that $OPT(G, k) = \text{CLC}(G, k, F)$ and $\mathcal{W}$ be a G/F-witness structure of G. Since we are working with a minimization problem, to prove this lemma it is sufficient to find a solution for G' which is of size $|F|$. Recall that (X, Y) is a partition of $V(G)$ such that $G - X = G[Y]$ is a clique. Let Y' be the set of vertices that were marked by either of the marking schemes. In other words, (X, Y') is a partition of G' such that $G' - X = G'[Y]$ is a clique. We proceed as follows. At each step, we construct graph $G^\star$ from G by deleting one or more vertices of $Y \setminus Y'$. Simultaneously, we also construct a set of edges $F^\star$ from F by either replacing the existing edges by new ones or by simply adding extra edges to F. At any intermediate state,

we ensure that $G^\star / F^\star$ is a clique, and the number of edges in $F^\star$ is at most $|F|$. Let $F^\circ = F$ be an optimum solution for the input instance (G, k). *For notational convenience, we rename $G^\star$ to G and $F^\star$ to F at regular intervals but do not change F°.*

To obtain $G^\star$ and $F^\star$, we delete witness sets which are subsets of $Y \setminus Y'$ (Condition 3.1) and modify the ones which intersect with $Y \setminus Y'$. Every witness set of latter type intersects with Y' or X or both. We partition these non-trivial witness sets in $\mathcal{W}$ into two groups depending on whether the intersection with X is empty (Condition 3.2) or not (Condition 3.3). We first modify the witness sets that satisfy the least indexed condition. If there does not exist a witness set which satisfies either of these three conditions, then $Y \setminus Y'$ is an empty set, and the lemma is vacuously true.

CONDITION 3.1. *There exists a witness set $W(t)$ in $\mathcal{W}$ which is a subset of $Y \setminus Y'$.*

Construct $G^\star$ from G by deleting the witness sets $W(t)$ in $\mathcal{W}$. Let $F^\star$ be obtained from F by deleting those edges whose both the endpoints are in $W(t)$. Since the class of cliques is closed under vertex deletion, $G^\star / F^\star$ is a clique, and as we only deleted edges from F, we have $|F^*| \leq |F|$. We repeat this process until there is no witness set that satisfies Condition 3.1.

> At this stage we rename $G^\star$ to G and $F^\star$ to F.

CONDITION 3.2. *There exists a witness set $W(t)$ in $\mathcal{W}$ which contains vertices from $Y \setminus Y'$ but does not intersect X.*

Since $W(t)$ is not contained in $Y \setminus Y'$ and $W(t) \cap X$ is empty it must intersect with Y'. See Figure 2. Let y_4 and y_5 be vertices in $W(t) \cap Y'$ and $W(t) \cap (Y \setminus Y')$, respectively. Let $W(t_1)$, different from $W(t)$, be a witness set which intersects Y'. Since Y' is large and non-empty, such a witness set exists. Let y_6 be a vertex in the set $W(t_1) \cap Y'$. Consider the witness sets $W(t), W(t_1)$ and vertex y_5 in $W(t)$ in graph G. Lemma 3.2 implies that these witness sets satisfy the premise of Observation 2.6. This implies $\mathcal{W}^\star$ is a clique witness structure of $G - \{y_5\}$, where $\mathcal{W}^\star$ is obtained from $\mathcal{W}$ by removing $W(t), W(t_1)$ and adding $(W(t) \cup W(t_1)) \setminus \{y_5\}$. This corresponds to replacing an edge in F which was incident to y_5 with the one across $W(t)$ and $W(t_1)$. For example, in Figure 2, we replace edge $y_4 y_5$ in the set F with an edge $y_4 y_6$ to obtain a solution for $G - \{y_5\}$. An edge in F has been replaced with another edge and one vertex in $Y \setminus Y'$ is deleted. The size of $F^\star$ is the same as that of F and $G^\star / F^\star$ is a clique. We repeat this process until there is no witness set which satisfies Condition 3.2.

> At this stage we rename $G^\star$ to G and $F^\star$ to F.

CONDITION 3.3. *There exists a witness set $W(t)$ in $\mathcal{W}$ which contains vertices from $Y \setminus Y'$ and intersects X.*

Let y be a vertex in $W(t) \cap (Y \setminus Y')$, X_t be the set of vertices in $W(t) \cap X$ which are adjacent to y via edges in F, and Q_t be the set of vertices in $W(t) \cap Y$ which are adjacent to y via edges in F. We find a *substitute* for y in Y'. If the set X_t is empty then the vertex y is adjacent only with the vertices of Y, in this case the edges incident to y can be replaced as mentioned in the Condition 3.2. Assume that X_t is non-empty. For every vertex x in X_t the set $\{x\}$ is considered by Marking Scheme 3.1. Since y is adjacent to every vertex x in X_t, the set $M_1(\{x\})$ is non-empty. As y is in $Y \setminus Y'$, and hence unmarked, for every x in X_t, there is a vertex in $M_1(\{x\})$, say y_x, different from y which has been marked. We construct $F^\star$ from F by the following operation: For every vertex x in X_t, replace the edge xy in F by xy_x. Fix a vertex x_o in X_t, and for every vertex u in Q_t, replace the edge uy in F with uy_{x_o}. Since we are replacing a set of edges in F with another set of edges of same size we have $|F^\star| \leq |F|$. (For example, in Figure 2, $X_t = W_1$ and $Q_t = \{y_7\}$. Edges yx_1, yy_7 are replaced by $x_1 y_1, y_1 y_7$ resp.) We argue that if $G^\star$ is obtained from G by removing y, then $G^\star / F^\star$ is a clique.

We argue that contracting edges in $F^\star$ partitions $W(t)$ into $|X_t| + |Q_t|$ many parts and merges each part with some witness set in $\mathcal{W} \setminus \{W(t)\}$. Recall that F contains a spanning tree of graph
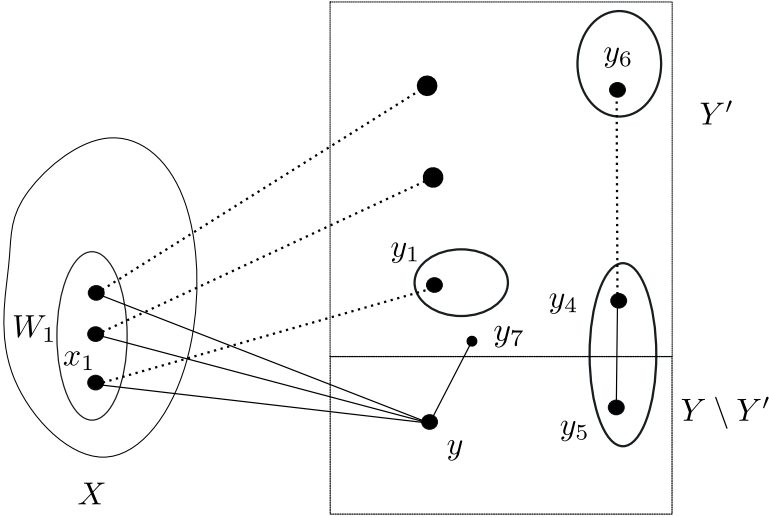
Fig. 2. Straight lines (e.g. $y_4 y_5$) represent edges in original solution F. Dotted lines (e.g. $y_4 y_6$) represents edges which are replaced for some edges in F. Please refer to the proof of Lemma 3.4.

$G[W(t)]$. Let T be a spanning tree of $G[W(t)]$ such that $E(T) \subseteq F$ and T contains all edges in F that are incident on y. It is easy to see that such a spanning tree exists. Let y be the root of tree T. For every z in $X_t \cup Q_t$, let $W'(z)$ be the set of vertices in the subtree of T rooted at z. As $V(T) = W(t)$, set $\{W'(z) | z \in X_t \cup Q_t\}$ is a partition of $W(t) \setminus \{y\}$. For every x in X_t, let $W(y_x)$ be the witness set in $\mathcal{W}$ containing the vertex y_x. For every x in $X_t \setminus \{x_o\}$, let $W^\star(y_x)$ be the set $W(y_x) \cup W'(x)$. Let $W^\star(y_{x_o})$ be the set $W(y_{x_o}) \cup W'(x_o) \cup \bigcup_{y'} W'(y')$ for all y' in Q_t. We obtain $\mathcal{W}^\star$ from $\mathcal{W}$ by removing $W(t)$ and $W(y_x)$ for every x in X_t, and adding the sets $W^\star(y_x)$ for every x in X_t. Since $W^\star(y_x)$ contains the set $W(y_x)$ which was adjacent to every witness set in $\mathcal{W}$, $W^\star(y_x)$ will be adjacent with every witness set in $\mathcal{W}^\star$. We repeat this process until there is no witness set that satisfies this condition.

Any vertex in $Y \setminus Y'$ must be a part of some witness set in $\mathcal{W}$, and any witness set in $\mathcal{W}$ satisfies at least one of the above conditions. If there are no witness sets that satisfy these conditions, then $Y \setminus Y'$ is empty. This implies $G^\star = G'$ and there exists a solution $F^\star$ of size at most $|F^\circ|$. This concludes the proof of the lemma. □

We are now in a position to prove the following lemma.

LEMMA 3.5. *Reduction Rule 3.2, along with a solution lifting algorithm, is an* $(1 + \epsilon)$-*reduction rule.*

PROOF. Let (G', k) be the instance returned by Reduction Rule 3.2 when applied on an instance (G, k). We present a solution lifting algorithm as follows. For a solution F' for (G, k) if $\text{CLC}(G', k, F') = k + 1$, then the solution lifting algorithm returns a spanning tree F of G (a trivial solution) as solution for (G, k). In this case, $\text{CLC}(G, k, F) = \text{CLC}(G', k, F')$. If $\text{CLC}(G', k, F') \leq k$, then the size of F' is at most k and G'/F' is a clique. Solution lifting algorithm uses Lemma 3.3 to construct a solution F for (G, k) such that cardinality of F is at most $(1 + \epsilon) \cdot |F'|$. In this case, $\text{CLC}(G, k, F) \leq (1 + \epsilon) \cdot \text{CLC}(G', k, F')$. Hence, there exists a solution lifting algorithm which given a solution F' for (G', k') returns a solution F for (G, k) such that $\text{CLC}(G, k, F) \leq (1 + \epsilon) \cdot \text{CLC}(G', k, F')$.

If $\text{OPT}(G, k) \leq k$, then by Lemma 3.4, $\text{OPT}(G', k) \leq \text{OPT}(G, k)$. If $\text{OPT}(G, k) = k + 1$ then $\text{OPT}(G', k) \leq k + 1 = \text{OPT}(G, k)$. Hence in either case, $\text{OPT}(G', k) \leq \text{OPT}(G, k)$.

Combining the two inequalities, we get $\frac{\text{CLC}(G, k, F)}{\text{OPT}(G, k)} \leq \frac{(1 + \epsilon) \cdot \text{CLC}(G', k, F')}{\text{OPT}(G', k)}$. This implies that if F' is a factor c-approximate solution for (G', k) then F is a factor $(c \cdot (1 + \epsilon))$-approximate solution for (G, k). This concludes the proof. □

We are now in a position to present the main result of this section.

THEOREM 1.1. *For any $\epsilon > 0$, CLIQUE CONTRACTION parameterized by the size of solution k, admits a time efficient $(1 + \epsilon)$-approximate polynomial kernel with $O(k^{d+1})$ vertices, where $d = \lceil \frac{1}{\epsilon} \rceil$.*

PROOF. For a given instance (G, k) with $|V(G)| \geq k + 3$, a kernelization algorithm applies the Reduction Rule 3.1. If it returns a trivial instance, then the statement is vacuously true. If it does not return a trivial instance, then the algorithm partitions $V(G)$ into two sets (X, Y) such that $G - X = G[Y]$ is a clique and size of X is at most $4k$. Then the algorithm applies the Reduction Rule 3.2 on the instance (G, k) with the partition (X, Y) and the integer $d = \lceil \frac{1}{\epsilon} \rceil$. The algorithm returns the reduced instance as $(1 + \epsilon)$-lossy kernel for (G, k).

The correctness of the algorithm follows from Lemma 3.1 and Lemma 3.5 combined with the fact that Reduction Rule 3.2 is applied at most once. By Observation 2.1, Reduction Rule 3.1 can be applied in polynomial time. The size of the instance returned by Reduction Rule 3.2 is at most $O((4k)^d \cdot (2k + 1) + 4k) = O(k^{d+1})$. Reduction Rule 3.2 can be applied in time $n^{O(1)}$ if the number of vertices in (G, k) is more than $O(k^{d+1})$. □

4 LOSSY KERNEL FOR SPLIT CONTRACTION

In this section, we present a lossy kernel for SPLIT CONTRACTION. We start by defining a natural optimization version of the problem.

$$\text{SPC}(G, k, F) = \begin{cases} \min\{|F|, k + 1\} & \text{if } G/F \text{ is a split graph} \\ \infty & \text{otherwise.} \end{cases}$$

We assume that the input graph is *connected and justify this assumption at the end.* If the number of vertices in the input graph is at most $k + 3$, then we return the same instance as a kernel for the given problem. Thus we only consider inputs that have at least $k + 3$ vertices. By the definition of optimization problem, for any set of edges F if G/F is a split graph then the maximum value of $\text{SPC}(G, k, F)$ is $k + 1$. Hence, any spanning tree of G is a solution of cost $k + 1$. We call it a *trivial solution* for the given instance. Consider an instance $(C_5, 1)$ where C_5 is a cycle on five vertices. One needs to contract at least two edges to convert C_5 into a split graph. We say $(C_5, 1)$ a *trivial no instance* for SPLIT CONTRACTION.

We start with a reduction rule, which says that if the minimum number of vertices that need to be deleted from an input graph to obtain a split graph is large, then we can return a trivial instance as a lossy kernel.

REDUCTION RULE 4.1. *Given an instance (G, k), apply the algorithm mentioned in Observation 2.2 to find a set S such that $G - S$ is a split graph. If $|S| > 10k$ then return $(C_5, 1)$.*

LEMMA 4.1. *Reduction Rule 4.1 is a 1-reduction rule.*

PROOF. Let (G, k) be an instance such that Reduction Rule 4.1 returns $(C_5, 1)$ when applied on it. Solution lifting algorithm returns a spanning tree F of G.

For a set of edges F', if C_5/F' is a split graph then F' contains at least two edges. This implies $\text{OPT}(C_5, 1) = 2$.

Since a factor 5-approximate algorithm returns a set of size strictly more than $10k$, for any S' of size at most $2k$, $G - S'$ is not a split graph. But by Observation 2.5 if G is k-contractible to a split graph then G can be converted into a split graph by deleting at most $2k$ vertices. Hence, for any set of edges $F^\star$, if $G/F^\star$ is a split graph, then the size of $F^\star$ is at least $k + 1$. This implies that $\text{OPT}(G, k) = k + 1$.

Combining these values, we get $\frac{\text{SpC}(G,k,F)}{\text{OPT}(G,k)} = \frac{k+1}{k+1} = \frac{2}{2} = \frac{\text{SpC}(C_5,1,F')}{\text{OPT}(C_5,1)}$. This implies that if F' is a factor c-approximate solution for $(C_5, 1)$, then F is a factor c-approximate solution for (G, k). This concludes the proof. □

We consider an instance (G, k) for which Reduction Rule 4.1 does not return a trivial instance. This implies that for a given graph G, in polynomial time, one can find a partition (S, X, Y) of $V(G)$ such that $|S|$ is at most $10k$ and $G - S$ is a split graph with (X, Y) as its split partition. Recall that, our objective is to present a $(2 + \epsilon)$-approximate polynomial kernel for given $\epsilon > 0$. Fix $\alpha = (2 + \epsilon)/2$. Find a smallest integer d such that $\frac{d+1}{d} \le \alpha$. In other words, fix $d = \lceil \frac{1}{\alpha - 1} \rceil$. If the number of vertices in the graph is at most $O(k^{d^2+d+1})$, then the algorithm returns the original graph as a lossy kernel of the desired size. Hence, without loss of generality, we assume that the number of vertices is at least $\Omega(k^{d^2+d+1})$.

Given an instance (G, k), a partition (S, X, Y) of $V(G)$, and an integer d, we mark some vertices in Y using the following two marking schemes.

MARKING SCHEME 4.1. *For a subset A of S, let $M_{NY}(A)$ be the set of neighbors of A in Y. For every subset A of S whose size is at most d, mark $k + 2$ vertices in $M_{NY}(A)$.*

Formally, $M_{NY}(A) = \{y \in Y | y \in N(A)\}$. If the number of vertices in $M_{NY}(A)$ is at most $k + 2$, then the marking scheme marks all vertices in M_{NY}, else it arbitrarily chooses $k + 2$ vertices and marks them.

MARKING SCHEME 4.2. *For a subset A of S, let $M_{CY}(A)$ be the set of vertices in Y whose neighborhood contains A. For every subset A of S whose size is at most d, mark a vertex in $M_{CY}(A)$. Marking scheme prefers a vertex with highest degree.*

Formally, $M_{CY}(A) = \{y \in Y | A \subseteq N(y)\}$. If $M_{CY}(A)$ is empty, then marking scheme does not mark any vertex, otherwise it picks a vertex with the highest degree.

Let Y' be the set of vertices of Y that have been marked by the Marking Schemes 4.1 or 4.2. Using set S and marked vertices in Y, Marking Schemes 4.3 and 4.4 marks some vertices in X. We remark that these two schemes are similar to Marking Schemes 3.1 and 3.2

MARKING SCHEME 4.3. *For a subset A of $S \cup Y'$, let $M_{CX}(A)$ be the set of vertices in X whose neighborhood contains A. For every subset A of $S \cup Y'$ whose size is at most d, mark two vertices in $M_{CX}(A)$.*

Formally, $M_{CX}(A) = \{x \in X | A \subseteq N(x)\}$. If $M_{CX}(A)$ is empty, then the marking scheme does not mark any vertex, and if it has only one vertex, then the marking scheme marks that vertex. If it has at least two vertices, then the marking scheme arbitrarily chooses two vertices and marks them.

MARKING SCHEME 4.4. *For a subset A of $S \cup Y'$, let $M_{NX}(A)$ be the set of vertices in X whose neighborhood do not intersect A. For every subset A of $S \cup Y'$ whose size is at most d, mark $2k + 2$ vertices in $M_{NX}(A)$.*

Formally, $M_{NX}(A) = \{x \in X | N(x) \cap A = \emptyset\}$. If the number of vertices in $M_{NX}(A)$ is at most $2k + 2$, then the marking scheme marks all vertices in $M_{NX}(A)$. If the number is greater than $2k + 2$, then the marking scheme arbitrarily chooses $2k + 2$ vertices and marks them.

REDUCTION RULE 4.2. *For a given instance (G, k), a partition (S, X, Y) of $V(G)$, and an integer d, apply Marking Schemes 4.1 to 4.4. Let G' be the graph obtained from G by deleting unmarked vertices of X and Y. Return the instance (G', k).*

The number of vertices in Y marked by Marking Schemes 4.1 and 4.2 is at most $O(|S|^{d+1})$. This implies that the total number of vertices marked by these four marking schemes is at most $O(|S|^{d^2+d+1})$. Above reduction rule can be applied in time $|S|^{d^2+d+1} \cdot |V(G)|^{O(1)} = |V(G)|^{O(1)}$ as $|S|$ is at most $10k$ and number of vertices in G is at least $O(k^{d^2+d+1})$. Note that G' is an induced subgraph of G, and hence $G' - S$ is a split graph with (X', Y') as its split partition. We first prove the following lemma which is similar to Lemma 3.2.

LEMMA 4.2. *Consider instance (G, k) of SPLIT CONTRACTION. Let Y' be the set of vertices marked by Marking Schemes 4.1 to 4.4 for some positive integer d. For any subset Z'' of $(X \setminus X') \cup (Y \setminus Y')$, let G'' be the graph obtained from G by deleting Z''. Then, G'' is connected.*

PROOF. Recall that, by our assumption, G is connected and X is a clique in G. Hence, for every vertex in $S \cup Y$, there exists a path from it to some vertex in X. By the construction of G'', $(S, X \setminus Z'', Y \setminus Z'')$ forms a partition of $V(G'')$ and $X \setminus Z''$ is a clique in G''. To prove that G'' is connected, it is sufficient to prove that for every vertex in $S \cup (Y \setminus Z'')$, there exists a path from it to a vertex in $X \setminus Z''$ in G.

We first prove that every vertex in $Y \setminus Z''$ has a path from it to a vertex in $X \setminus Z''$. Fix an arbitrary vertex, say y, in $Y \setminus Z''$. Consider a path P from y to x in G, where x is some vertex in X. Without loss of generality, we can assume that x is the only vertex in $V(P) \cap X$. We argue that there exists a path, say P_1, from y to a vertex in $X \setminus Z''$. If x is in $X \setminus Z''$ then $P_1 = P$ is a desired path. Consider the case when x is in Z''. Let w be the vertex in $V(P)$ which is adjacent with x. Note that w is either in S or in Y. It may be the same as y. As Marking Scheme 4.3 considered all subsets of size at most d in $S \cup (Y \setminus Z'')$, it considered the singleton set $\{w\}$. As w is adjacent with x, we have $\{w\} \subseteq N(x)$. Since x is in Z'', and hence unmarked, there exists a vertex, say x_1, in X which has been marked by Marking Scheme 4.3. Consider the path P_1 obtained from P by deleting vertex x (and hence edge wx) and adding vertex x_1 with edge wx_1. This is a desired path from y to a vertex in $X \setminus Z''$. As y is an arbitrary vertex in $Y \setminus Z''$, this statement is true for any vertex in $Y \setminus Z''$ and hence G'' is connected.

Now, it is sufficient to argue that for every vertex in S, there exists a path from it to a vertex in $(X \cup Y) \setminus Z''$. Fix an arbitrary vertex, say s, in S. Consider a path P from s to u in G, where u is some vertex in $X \cup Y$. As G is connected, such a path exists. Without loss of generality, we can assume that u is the only vertex in $V(P) \cap (X \cup Y)$. We argue that there exists a path, say P_1, from s to a vertex in $(X \cup Y) \setminus Z''$. If u is in $(X \cup Y) \setminus Z''$ then $P_1 = P$ is a desired path. Consider the case when u is in Z''. Let s_0 be the vertex in $V(P)$ which is adjacent with u. Note that s_0 is in S and it may be the same as s. As Marking Scheme 4.1 and 4.3 considered all subsets of size at most d in S, it considered the singleton set $\{s_0\}$. As s_0 is adjacent with u, we have $\{s_0\} \subseteq N(u)$. Since u is in Z'', and hence unmarked, there exists a vertex, say u_1, in $(X \cup Y) \setminus Z''$ which has been marked by Marking Scheme 4.1 or 4.3. Consider a path P_1 obtained from P by deleting vertex u (and hence edge s_0u) and adding the vertex u_1 with edge s_0u_1. This is a desired path from s to a vertex in $(X \cup Y) \setminus Z''$. As s is an arbitrary vertex in S, this statement is true for any vertex in S.

Hence, there exists a path from every vertex in $S \cup (Y \setminus Z'')$ to a vertex in $X \setminus Z''$ in G''. As $X \setminus Z''$ is a clique in G'', we can conclude that G'' is connected.   □

As in case of CLIQUE CONTRACTION, we will iteratively remove vertices from $(X \setminus X') \cup (Y \setminus Y')$, and Lemma 4.2 ensures that the graph at each step remains connected.
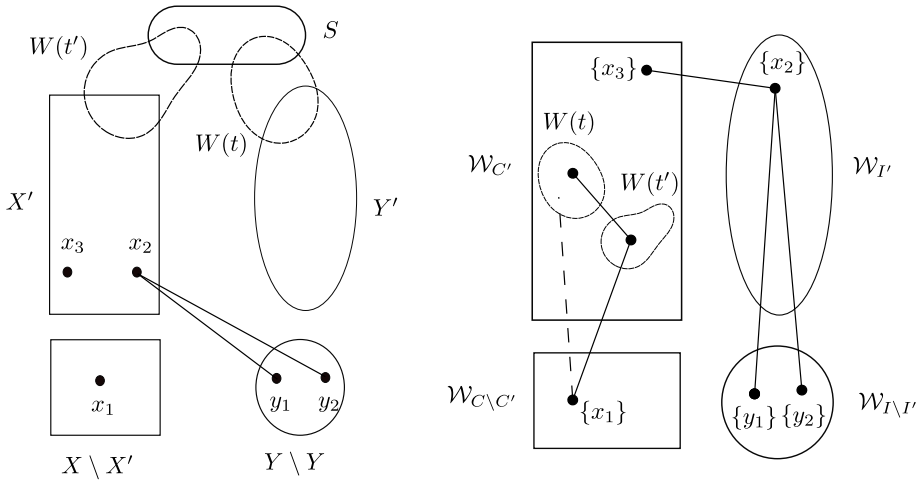
Fig. 3. Left hand side figure shows partition (S, X, Y) of G where as right hand side figure shows witness sets in G and G'. We need to modify witness sets in $\mathcal{W}_I$ which are adjacent with each other (e.g. $\{y_1\}$ and $\{x_2\}$) and witness sets in $\mathcal{W}_C$ which are not adjacent with each other (e.g. $W(t)$ and $\{x_1\}$). Please refer to Lemma 4.3.

To avoid corner cases, we need to ensure that whenever $X \setminus X'$ is non-empty, there is at least one witness set, which contains a vertex in X'. We ensure that by marking a few additional vertices in X.

REMARK 4.1. *Mark any $2k + 2$ vertices in X.*

Note that we can not infer anything about the adjacency of these vertices with vertices in $S \cup Y$. We use these vertices only to add certain edges, which are entirely contained in X.

In Lemma 4.3, we argue that given a solution for (G', k), we can construct a solution of almost the same size for (G, k).

LEMMA 4.3. *Let (G', k) be the instance returned by Reduction Rule 4.2 when applied on the instance (G, k). If there exists a set of edges of size at most k, say F', such that G'/F' is a split graph then there exists a set of edges F such that G/F is a split graph and cardinality of F is at most $\alpha \cdot |F'| + 1$.*

PROOF. If no vertex in $X \cup Y$ has been deleted, then G' and G are identical graphs, and the statement is true. We assume that at least one vertex from $X \cup Y$ has been deleted. Recall that X', Y' is the set of vertices in X, Y, respectively, that have been marked. It is easy to see that (S, X', Y') is a partition of $V(G')$ such that $G' - S$ is a split graph with (X', Y') as one of its split partition.

Let $\mathcal{W}'$ be a G'/F'-witness structure of G'. By Lemma 4.2, G' is connected and without loss of generality we assume that edges F' satisfies the three properties mentioned in Observation 2.7. Let (C', I') be the split partition of G'/F' mentioned in the third property in Observation 2.7. Let $\mathcal{W}_{C'}$ (resp. $\mathcal{W}_{I'}$) be the collection of witness set in $\mathcal{W}'$ which correspond to vertices in C' (resp. I'). We intentionally name a subset of $\mathcal{W}'$ with $\mathcal{W}_{C'}$ (instead of $\mathcal{W}'_{C'}$) as it simplifies notations in remaining proof. Note that any two witness sets in $\mathcal{W}_{C'}$ are adjacent with each other in G' and no two witness sets in $\mathcal{W}_{I'}$ are adjacent with each other in G'. By Observation 2.7, all non-trivial witness sets in $\mathcal{W}'$ are contained in $\mathcal{W}_{C'}$.

We start constructing witness structure $\mathcal{W}$ and set a of edges F of G from $\mathcal{W}'$ and F' as follows. For every vertex u in $(X \setminus X') \cup (Y \setminus Y')$, add singleton witness sets $\{u\}$ to $\mathcal{W}'$. Initialize F to F'. Let $\mathcal{W}_{C \setminus C'}$ be the set of newly added singleton witness sets that correspond to the vertices in $X \setminus X'$ and let $\mathcal{W}_{I \setminus I'}$ be the set of newly added singleton witness sets that correspond to the vertices in $Y \setminus Y'$.

In the remaining proof, we argue that we can carefully add some edges in F' such that following two conditions are satisfied. (i) Any two witness sets in $\mathcal{W}_{C'} \cup \mathcal{W}_{C \setminus C'}$ are adjacent with each other in G (ii) No two witness sets in $\mathcal{W}_{I'} \cup \mathcal{W}_{I \setminus I'}$ are adjacent with each other in G. To ensure condition (i), we might have to add one extra edge for d edges present in F'. **This addition of edges introduces the multiplicative factor of α in the upper bound for the size of F in terms of F'. To ensure condition (ii), we might have to contract an edge outside F'. This brings an additive factor of one.**

Let $\mathcal{W}^*$ be the collection of witness sets in $\mathcal{W}_{C'}$ which violates Condition (i). In other words, $\mathcal{W}^*$ is the collection of witness set $W(t)$ in $\mathcal{W}_{C'}$ such that there exists a (singleton) witness set $\{x\}$ in $\mathcal{W}_{C \setminus C'}$ which is not adjacent with $W(t)$. See Figure 3. (For example, here witness set $\{x_1\}$ is not adjacent with $W(t)$.) We argue that every witness set in $\mathcal{W}^*$ has at least $d + 1$ vertices. For the sake of contradiction, assume that there exists a witness set $W(t)$ in $\mathcal{W}^*$ which contains at most d vertices. Let $\{x_1\}$ be a singleton set in $\mathcal{W}_{C \setminus C'}$ which is not adjacent with $W(t)$. Since $G[X]$ induces a clique in G, witness set $W(t)$ is contained in $S \cup Y'$. Since Marking Scheme 4.4 iterated over all sets of size at most d, it also considered $W(t)$ while marking. Note that x_1 is contained in $M_{NX}(W(t))$, a set of non-neighbors of $W(t)$ in X. As x_1 is unmarked, there are $2k + 2$ vertices in $M_{NX}(W(t))$ that have been marked. All these marked vertices are in G'. Since the cardinality of F' is at most k, the number of vertices in $V(F')$ is at most $2k$. This implies that at least two marked vertices in $M_{NX}(W(t))$ remain as singleton witness sets. Since these two witness sets are adjacent to each other, at least one of these sets in contained in $\mathcal{W}_{C'}$. This contradicts the fact that any two witness sets in $\mathcal{W}_{C'}$ are adjacent to each other. Hence our assumption is wrong and $W(t)$ has at least $d + 1$ vertices.

Fix a witness set, say $W(t')$, in $\mathcal{W}_{C'}$ which intersects with X'. By Remark 4.1, X' is non-empty and such set exists. We note that $W(t')$ is adjacent with every (singleton) witness set in $\mathcal{W}_{C \setminus C'}$. For every witness set $W(t)$ in $\mathcal{W}^*$, we add an edge between $W(t)$ and $W(t')$ to the set F'. Equivalently, we construct a new witness set by taking the union of $W(t')$ and all the witness sets $W(t)$ in $\mathcal{W}^*$.

We now argue the size bound of F. Note that we have added one extra edge for every witness set $W(t)$ in $\mathcal{W}^*$. As every witness set of $\mathcal{W}^*$ has at least $d + 1$ vertices, we have added one extra edge for at least d edges in the solution F. Moreover, since witness sets in $\mathcal{W}^*$ are vertex disjoint, no edge in F' can be part of two witness sets. This implies that the number of edges in F is at most $\frac{d+1}{d}|F'| \leq \alpha \cdot |F'|$.

We now consider Condition (ii). Let $\mathcal{W}^*$ be the collection of witness sets in $\mathcal{W}_{I'}$ which violates Condition (ii). In other words, $\mathcal{W}^*$ is the collection of witness set $W(t)$ in $\mathcal{W}_{I'}$ such that there exists a (singleton) witness set $\{y\}$ in $\mathcal{W}_{I \setminus I'}$ which is adjacent with $W(t)$. Since Y is an independent set in G, any witness set $W(t)$ in $\mathcal{W}^*$ intersects with either X' or S. We consider two cases depending on whether $W(t)$ intersects X' or not. We argue that in the first case, we can add an extra edge to F' and avoid all such cases, while the second case can not occur.

Consider a witness set $W(t)$ in I' which intersects with X'. Let a be the vertex in I' and $W_a = W(t)$ be the witness set corresponding to it. By Remark 4.1, there exists vertex b in G'/F' which is adjacent to a. Let W_b be the witness set in $\mathcal{W}'$ corresponding to b. Remove W_b, W_a from $\mathcal{W}'$ and add $W_a \cup W_b$ as a witness set to $\mathcal{W}'$ (or more specifically to $\mathcal{W}_{C'}$). Since X' is a clique, at most one vertex from X' can be part of witness set in $\mathcal{W}_{I'}$. Hence there is at most one such vertex in I'. Since edge across

W_a, W_b is not in F', this operation adds one extra edge in F'. (For example, edge x_2x_3 in Figure 3). Hence, in order to make sure that no witness set violates Condition (*i*) and (*ii*), we have added edges to F' to obtain F such that the size is at most $\alpha|F'| + 1$.

Now, consider the second case. Assume that there exists a witness $W(t)$ in $\mathcal{W}^*$ which does not intersect with X'. This implies that $W(t)$ is contained in S. Let y_1 be a singleton witness set in $\mathcal{W}_{I \setminus I'}$ which is adjacent with $W(t)$. By Observation 2.7, we know that $W(t)$ is a singleton witness set and is contained in S. Hence set $A = W(t)$ has been considered by Marking Scheme 4.1. Note that y_1 is contained in $M_{NY}(A)$, a set of neighbors of A in I. Since y_1 is unmarked, there are $k + 2$ vertices in $M_{NY}(A)$ that have been marked. All these marked vertices are in G'. Since Y' is an independent set in G', at most k vertices in Y' can be incident on solution edges. Only these vertices can be part of $\mathcal{W}_{C'}$. There can be at most one vertex which is a singleton witness set in $\mathcal{W}_{C'}$. Hence there exists at least one singleton witness set in $\mathcal{W}_{I'}$ which is adjacent with $W(t)$. This contradicts the fact that no two witness sets in $\mathcal{W}_{I'}$ are adjacent to each other. Hence our assumption is wrong, and no such witness structure exists in $\mathcal{W}^*$. This concludes the proof of the lemma. □

In the following lemma, we argue that the value of the optimum solution for reduced instance can be upper bounded by the value of the optimum solution for the original instance.

LEMMA 4.4. *Let (G', k) be the instance returned by Reduction Rule 4.2 when applied on an instance (G, k). If $OPT(G, k) \leq k$ then $OPT(G', k) \leq 2 \cdot OPT(G, k)$.*

PROOF. Let F be a set of at most k edges in G such that $OPT(G, k) = \text{SpC}(G, k, F)$. Since we are working with a minimization problem, to prove the lemma, it is sufficient to find a solution for G', which is of size at most $2 \cdot |F|$. Recall that (S, X, Y) is a partition of $V(G)$ such that $G - S$ is a split graph with (X, Y) as split partition where X is a clique, and Y is an independent set. The set of vertices that have been marked in Y is denoted by Y', and the set of vertices that have been marked in X is denoted by X'. By our assumption, the input graph G is connected. Without loss of generality, we can assume that F satisfies three properties mentioned in Observation 2.7. Let (C, I) be the split partition of G/F mentioned in the third property in Observation 2.7. Let $\mathcal{W}$ be a G/F-witness structure of G. Let $\mathcal{W}_C$ (resp. $\mathcal{W}_I$) be the collection of witness sets in $\mathcal{W}$ which correspond to vertices in C (resp. I). Note that any two witness sets in $\mathcal{W}_C$ are adjacent with each other in G, and no two witness sets in $\mathcal{W}_I$ are adjacent with each other in G. By Observation 2.7, all non-trivial witness sets in $\mathcal{W}$ are contained in $\mathcal{W}_C$.

At each step, we construct the graph G^* from G by deleting one or more vertices in $(X \setminus X') \cup (Y \setminus Y')$. By Lemma 4.2, G^* is connected. **We also construct a set of edges F^* from F by replacing existing edges and/or adding extra edges to F.** In terms of witness sets, we delete witness sets that are subsets of $(X \setminus X') \cup (Y \setminus Y')$ and modify the ones that intersect $(X \setminus X') \cup (Y \setminus Y')$. After the modification, we represent the witness sets corresponding to vertices in the clique as $\mathcal{W}_C^*$ and the independent set as $\mathcal{W}_I^*$. At any point, we ensure that any two witness sets in $\mathcal{W}_C^*$ are adjacent to each other, and any two witness sets in $\mathcal{W}_I^*$ are not adjacent to each other. This implies that at any intermediate state, G^*/F^* is a split graph. We modify F to obtain F^* such that the number of edges in F^* is at most $2 \cdot |F^\circ|$, where $F^\circ = F$ be an optimum solution for the original instance (G, k). For notational convenience, we rename G^* to G and F^* to F at regular intervals but do not change F°.

Since the class of split graphs is closed under vertex deletion, we can delete all witness sets, which are entirely contained in $(X \setminus X') \cup (Y \setminus Y')$. Suppose that there exists a witness set $W(t)$ in $\mathcal{W}$ which is a subset of $(X \setminus X') \cup (Y \setminus Y')$, construct G^* from G by deleting witness set $W(t)$ in $\mathcal{W}$. Delete the edges corresponding to spanning tree of $G[W(t)]$ from F to obtain F^*. We repeat

this process until there is no witness set that satisfies this condition. After exhaustively applying this process, we have $|F^*| \leq |F|$.

<div style="border:1px solid black; padding:8px; text-align:center;">
At this stage we rename $G^\star$ to G and $F^\star$ to F.
</div>

Note that at this stage, there is no witness set in $\mathcal{W}_I$ which contains vertex in $(X \setminus X') \cup (Y \setminus Y')$. Hence, we do not need to modify witness sets in $\mathcal{W}_I$. In all the conditions mentioned below, the modification is done on non-trivial witness sets in $\mathcal{W}_C$ only. These modifications do not affect the independent property of witness sets in $\mathcal{W}_I$. So, to prove that the modified witness structure obtained corresponds to a split graph, it is enough to show that the witness sets in $\mathcal{W}_C^*$ are connected, and any two of them are adjacent to each other.

We partition all non-trivial witness sets in $\mathcal{W}_C$ with respect to their intersection with sets S, X, Y. For a non-trivial witness set $W(t)$, we denote its intersection with S, X, Y, respectively, using an ordered tuple $(i; j; k)$ where integers i, j, k take 0 or 1. If $W(t)$ intersects with Z in $\{S, X, Y\}$, then we assign the corresponding integer to 1 and 0 otherwise. Since (S, X, Y) is a partition of $V(G)$, witness sets in $\mathcal{W}_C$ can be partitioned into seven parts (excluding the trivial $(0; 0; 0)$ case). Note that since Y is an independent set in G, no non-trivial witness set can contain only vertices in Y. This implies that there is no non-trivial witness set in the partition of $\mathcal{W}_C$ corresponding to $(0; 0; 1)$. Consider a witness set that is entirely contained in S. Since we do not delete any vertex in S while creating G^* from G, this witness set remains unchanged throughout the process. Hence we do not consider witness sets in partition corresponding to $(1; 0; 0)$.

This implies we only need to modify witness sets in five partitions of $\mathcal{W}_C$. Each of these partitions can be further divided into subparts based on whether witness set intersects with $X \setminus X'$ or $Y \setminus Y'$ or both or none. We note that any witness set which does not intersect with $(X \setminus X') \cup (Y \setminus Y')$ is not affected; hence we only need to consider first three cases. We modify a witness set that satisfies the least indexed condition. If there does not exist any witness set which satisfies either of these conditions, then $(X \setminus X') \cup (Y \setminus Y')$ is an empty set, and the lemma is vacuously true. Hence in every case, we assume that witness set intersects $(X \setminus X') \cup (Y \setminus Y')$, and it is not entirely contained inside it.

Condition 1: [Partition $(0; 1; 0)$] There exists a witness set, say $W(t)$, which intersects with X but does not intersect with S or Y.

Since $W(t)$ is contained in set X, it intersects with $X \setminus X'$ but is not contained in it. Hence both the sets $W(t) \cap X'$ and $W(t) \cap (X \setminus X')$ are non empty. Let x_1 be a vertex in $W(t) \cap (X \setminus X')$. Fix a witness set, say $W(t^\#)$, in $\mathcal{W}_C$ which is different from $W(t)$ and intersects with X'. By Remark 4.1, X' is of size at least $2k + 2$ and hence such a set exists. Let $\mathcal{W}_C^*$ be the witness set obtained from $\mathcal{W}_C$ by removing $W(t), W(t^\#)$ and adding $(W(t) \setminus \{x_1\}) \cup W(t^\#)$. Note that in $\mathcal{W}_C$, $W(t), W(t^\#)$ and x_1 satisfy the premise of Observation 2.6. This implies that any two witness sets in $\mathcal{W}_C^*$ are adjacent with each other. Let F^* be the set of edges obtained from F by removing an edge incident on x_1 and adding an edge across $W(t)$ and $W(t^\#)$. Hence, if G^* is obtained from G by deleting x_1 then G^*/F^* is a split graph. Since we are deleting at least one edge from F and adding only one edge, we have $|F^*| \leq |F|$.

<div style="border:1px solid black; padding:8px; text-align:center;">
At this stage we rename $G^\star$ to G and $F^\star$ to F.
</div>

Condition 2: [Partition $(1; 1; 0)$] There exists a witness set, say $W(t)$, which intersects with S and X but does not intersect with Y. Since $W(t)$ does not intersect with Y, it intersects with $X \setminus X'$. Let x be a vertex in $W(t) \cap (X \setminus X')$; S_t be the set of vertices in $W(t) \cap S$ which are adjacent with x via edges in F and Q_t be the set of vertices in $W(t) \cap X$ which are adjacent with x via edges of F. We find a *substitute* for x in X'. (This condition is the same as that of Condition 3.3 in the proof of Lemma 3.4.) Note that if S_t is empty then Q_t must be non-empty and we replace the edges incident
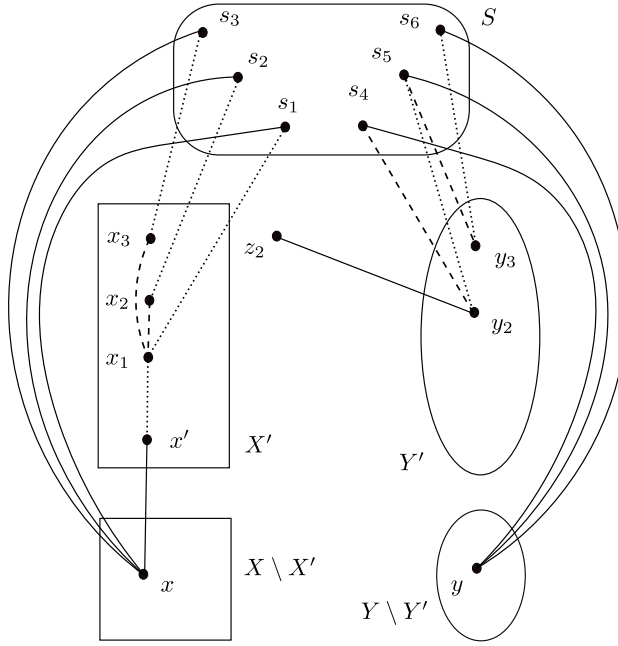
Fig. 4. Straight lines (e.g. xx') represent edges in original solution F. Dotted lines (e.g. x_1x') represents an edges replaced in F. Dashed lines (e.g. x_1x_2) represents extra edges added to F. Please refer to Conditions 2 and 3 in Lemma 4.4. Vertex z_2 can be in X or S.

on x similar to that of in Condition 1. Every vertex in S_t is considered by Marking Scheme 4.3. For every s in S_t, vertex x is contained in $M_{CX}(\{s\})$, the set of vertices in X whose neighborhood contains $\{s\}$. Since x is in $X \setminus X'$, and hence unmarked, there are two vertices in $M_{CX}(\{s\})$, say x_s^1, x_s^2, different from x which have been marked for each s in S_t. Since X is a clique in G, at most one of these two vertices are part of witness set contained in $\mathcal{W}_I$. Without loss of generality, assume that for every vertex s in S_t, vertex x_s^1 is contained in witness set contained in $\mathcal{W}_C$. We construct F^* from F by following operation: Arbitrarily fix a vertex s_0 in S_t. Remove the edge $s_0 x$ in F and add the edge $s_0 x_{s_0}^1$ and for every vertex s in $S_t \setminus \{s_0\}$, remove the edge sx in F and add two edges $sx_s^1, x_s^1 x_{s_0}^1$. For every vertex x' in Q_t replace the edge $x'x$ by $x'x_{s_0}^1$. (For example, in Figure 4, let $W(t) = \{x, x', s_1, s_2, s_3\}$ and x is in $X \setminus X'$. This implies that $S_t = \{s_1, s_2, s_3\}$ and $Q_t = \{x'\}$. Let s_1 be the vertex which is fixed arbitrarily. In this process, edge xs_3 is deleted and two edges x_3s_3, x_3x_1 are added to F.)

We argue that contracting edges in F^* merges $W(t) \setminus \{x\}$ into the witness set in $\mathcal{W}_C^*$ that contains x_s^1. Define $W := (W(t) \setminus \{x\}) \cup (\bigcup_{s \in S_t} W(x_s^1))$ where $W(x_s^1)$ is the witness set in $\mathcal{W}$ containing the vertex x_s^1. Let $\mathcal{W}_C^*$ be the witness structure obtained from $\mathcal{W}_C$ by removing every witness set which intersects W and adding W. Since W contains $W(x_{s_0}^1)$, it is adjacent with every other witness set in $\mathcal{W}_C$ and hence in $\mathcal{W}_C^*$. This implies that any two witness sets in $\mathcal{W}_C^*$ are adjacent to each other. Hence, if G^* is obtained from G by deleting x, then G^*/F^* is a split graph. Since we are adding at most two edges only for deleted edges in F, we have $|F^*| \leq 2 \cdot |F|$.

> At this stage we rename $G^\star$ to G and $F^\star$ to F.

Condition 3: [Partition $(1; 0; 1)$] There exists a witness set, say $W(t)$, which intersects with S and Y but does not intersect with X.

Since $W(t)$ does not intersect with X, it intersects with $Y \setminus Y'$. Let y be a vertex in $W(t) \cap (Y \setminus Y')$; S_t be the set of vertices in $W(t) \cap S$ which are adjacent with y via edges in F. After removing all the edges incident on y we add some edges to ensure connectivity of vertices in S_t and some other to ensure adjacency among witness sets in $\mathcal{W}_C$. Arbitrarily fix a vertex s_0 in S_t. If $S_t \setminus \{s_0\}$ is an empty set then no edge needs to be included to ensure connectivity. For every vertex s in $S_t \setminus \{s_0\}$, set $\{s_0, s\}$ is considered by Marking Scheme 4.2. (For any $\alpha > 1$, d is at least two.) For every s in $S_t \setminus \{s_0\}$, vertex y is contained in $M_{CY}(\{s_0, s\})$, the set of common neighbors of $\{s_0, s\}$ in Y. Since y is in $Y \setminus Y'$, and hence unmarked, there is a vertex in $M_{CY}(\{s_0, s\})$ different from y which has been marked for each set $\{s_0, s\}$. For every vertex s in $S_t \setminus \{s_0\}$, let y_s be the marked vertex which is adjacent with s_0 and s. We construct F^* from F by following operation: Remove the edge $s_0 y$ and for every vertex s in $S_t \setminus \{s_0\}$, remove the edge sy in F and add two edges $sy_s, s_0 y_s$. (For example, in Figure 4, let $W(t) = \{y, s_4, s_5, s_6\}$ and y is in $Y \setminus Y'$. This implies that $S_t = \{s_4, s_5, s_6\}$. Let s_4 be the vertex which is fixed arbitrarily. In this process, edges $s_4 y, s_5 y$ are deleted and two edges $s_5 y_2, s_4 y_2$ are added to F.) Now we include additional edges to ensure adjacency among witness sets in $\mathcal{W}_C$. Towards this we first prove that there always exists a witness set $W(t^\#)$ different from $W(t)$, in $\mathcal{W}_C$ that is adjacent with either $W(t) \setminus \{y\}$ or with y_s for some s in $S_t \setminus \{s_o\}$. Suppose that there is no set $W(t^\#)$ that is adjacent with $W(t) \setminus \{y\}$ this implies that every witness set in $\mathcal{W}$ is adjacent with only vertex y in $W(t)$. As the size of X' is at least $2k + 2$ and y is adjacent with every witness set in $\mathcal{W} \setminus \{W(t)\}$ and every vertex in S_t, y is adjacent with at least $k + 2$ vertices. For a vertex s' in $S_t \setminus \{s_o\}$, let $y_{s'}$ be the vertex marked by Marking Scheme 4.2 while considering set $\{s_o, s'\}$. Note that Marking Scheme 4.2 preferred $y_{s'}$ over y. This implies that $y_{s'}$ is adjacent with at least $k + 2$ vertices. Hence, $y_{s'}$ has at least one neighbor outside $W(t)$. Note that $y_{s'}$ is not in the clique because $W(t) \setminus \{y\}$ did not have a neighbor in $\mathcal{W}_C$, this implies that the neighbor of $y_{s'}$ is in $\mathcal{W}_C$. Since there always exists another witness set $W(t^\#)$ that is adjacent with $(W(t) \setminus \{y\}) \cup \{y_s| \text{ for every } s \text{ in } S_t \setminus \{s_o\}\}$, we add the edge across $W(t), W(t^\#)$ in F.

Let W be the superset of $W(t) \setminus \{y\}$ which contains witness sets of all the newly added vertices to $W(t)$. Formally, $W = (W(t) \setminus \{y\}) \cup (\bigcup_{s \in S_t \setminus \{s_o\}} W(y_s))$ where $W(y_s)$ is the witness set containing y_s. Note that every vertex in S_t is connected with each other (as we have added edges $y_s s, y_s s_0$ for every s in $S_t \setminus \{s_o\}$) and W is adjacent with $W(t^\#)$. Let $\mathcal{W}_C^*$ be the witness structure obtained from $\mathcal{W}_C$ by removing every witness set which intersects with $W, W(t^\#)$ and adding $W \cup W(t^\#)$. Since $W \cup W(t^\#)$ contains $W(t^\#)$, which is adjacent with every other witness set in $\mathcal{W}_C$ and hence in $\mathcal{W}_C^*$. This implies that any two witness sets in $\mathcal{W}_C^*$ are adjacent with each other. Hence, if G^* is obtained from G by deleting y then G^*/F^* is a split graph. Since we are adding two edges of the form $y_s s, y_s s_0$ for every edge of the form sy in F, where $s \in S_t \setminus \{s_o\}$ and one edge across $W(t), W(t^\#)$ instead of $y s_o$ we have $|F^*| \le 2 \cdot |F|$.

> At this stage we rename $G^\star$ to G and $F^\star$ to F.

Condition 4: [Partition $(0; 1; 1)$] There exists a witness set, say $W(t)$, which intersects with X and Y but does not intersect with S.

We construct F^* from F in two steps, where the first step deletes the vertices of $Y \setminus Y'$, and the second step deletes the vertices of $X \setminus X'$. Suppose that there exists a vertex y in $W(t) \cap (Y \setminus Y')$. Fix a witness set, say $W(t^\#)$, in $\mathcal{W}_C$ which is different from $W(t)$ and intersects X'. By Remark 4.1, X' is of size at least $2k + 2$ and hence such set exists. Let $\mathcal{W}_C^*$ be the witness set obtained from $\mathcal{W}_C$ by removing $W(t), W(t^\#)$ and adding $W(t^\#) \cup (W(t) \setminus \{y\})$. Note that in $\mathcal{W}_C$, witness sets $W(t), W(t^\#)$

and y satisfy the premise of Observation 2.6. This implies that any two witness sets in $\mathcal{W}_C^*$ are adjacent with each other. Hence, if G^* is obtained from G by deleting y then G^*/F^* is a split graph. Since we are deleting at least one edge from F and adding only one edge, we have $|F^*| \leq |F|$. We repeat this process as long as there exists a witness set that satisfy Condition 4 and intersects $Y \setminus Y'$.

Consider a witness set $W(t)$ which satisfies Condition 4 and does not intersect with $Y \setminus Y'$. This implies $W(t)$ intersects with $X \setminus X'$ and every vertex in $W(t) \cap Y$ is contained in Y'. Consider a vertex x in $W(t) \cap (X \setminus X')$. If F does not contain any edges across $\{x\}$ and Y' then in $\mathcal{W}_C$, witness sets $W(t), W(t^{\#})$ and x satisfy the premise of Observation 2.6. We obtain the modified graph as mentioned in previous case. Suppose that there exist edges in F across $\{x\}$ and Y'. Let Y_t be the set of neighbors of x in Y' via edges in F. For every y in Y_t, the set $\{y\}$ is considered by Marking Scheme 4.3. Note that x is contained in the set $M_{CX}(\{y\})$, the set of vertices in X whose neighborhood contains $\{y\}$. Since x is in $X \setminus X'$, and hence unmarked, there are two vertices in $M_{CX}(\{y\})$, say x_y^1, x_y^2, different from x which have been marked for each y in Y_t. Since X is a clique in G, at most one of these two vertices are part of witness set contained in $\mathcal{W}_I$. Without loss of generality, assume that for every vertex y in Y_t, vertex x_y^1 is contained in the witness set in $\mathcal{W}_C$. We construct F^* from F by following operation (This step is similar modification as we did in Condition 2): Arbitrarily fix a vertex y_0 in Y_t. For every vertex y in Y_t, remove the edge yx in F and add two edges $yx_y^1, x_y^1 x_{y_0}^1$. Let $W = (W(t) \setminus \{x\}) \cup (\bigcup_{y \in Y_t} W(x_y^1))$ where $W(x_y^1)$ is the witness set in $\mathcal{W}$ containing x_y^1. We obtain the witness structure $\mathcal{W}_C^*$ from $\mathcal{W}_C$ by removing every witness set that intersects W and adding W. Since W contains $W(x_{y_0}^1)$, it is adjacent with every other witness set in $\mathcal{W}_C$ and hence in $\mathcal{W}_C^*$. This implies that any two witness sets in $\mathcal{W}_C^*$ are adjacent to each other. Hence, if G^* is obtained from G by deleting x, then G^*/F^* is a split graph. Since we are adding at most two edges $yx_y^1, x_y^1 x_{y_o}^1$ for one deleted edge of the form yx in F, we have $|F^*| \leq 2 \cdot |F|$.

> At this stage we rename $G^\star$ to G and $F^\star$ to F.

Condition 5: [Partition $(1; 1; 1)$] There exists a witness set, say $W(t)$, which intersects with S, X and Y.

We further divide this condition based on whether the intersection of $W(t)$ with $(Y \setminus Y')$ is empty (Condition 5(a)) or not (Condition 5(b))

Condition 5(a): Consider that $W(t) \cap (Y \setminus Y')$ is empty. Then there is at least one vertex in $W(t) \cap (X \setminus X')$. In this case, construction of G^* and F^* is similar to that of Condition 2. Instead of considering a subset of S in Condition 2, we consider a subset of $S \cup Y'$. Let x be a vertex in $W(t) \cap (X \setminus X')$; Z_t be the set of vertices in $W(t) \cap (S \cup Y')$ which are adjacent with x via edges in F and Q_t be the set of vertices in $W(t) \cap X'$ which are adjacent with x via edges of F. Note that every vertex in Z_t is considered by the Marking Scheme 4.3. For every vertex z in Z_t, vertex x is contained in $M_{CX}(\{z\})$, the set of vertices in X whose neighborhood contains $\{z\}$. Since x is in $X \setminus X'$, and hence unmarked, there are two vertices in $M_{CX}(\{z\})$, say x_z^1, x_z^2, different from x which have been marked for each z in Z_t. Since X is a clique in G, at most one of these two vertices are part of witness set contained in $\mathcal{W}_I$. Without loss of generality, assume that for every vertex z in Z_t, vertex x_z^1 is contained in witness set of $\mathcal{W}_C$. We construct F^* from F by following operation: Arbitrarily fix a vertex z_0 in Z_t. remove the edge $z_0 x$ and for every vertex z in $Z_t \setminus \{z_0\}$, remove the edge zx in F and add two edges $zx_z^1, x_z^1 x_{z_0}^1$. For every vertex x' in Q_t replace the edge $x'x$ by $x' x_{z_0}^1$.

Let W be the superset of $W(t) \setminus \{x\}$ which contains witness set containing all newly added vertices in $W(t)$. Formally, $W = (W(t) \setminus \{x\}) \cup (\bigcup_{z \in Z_t} W(x_z^1))$ where $W(x_z^1)$ is the witness set in $\mathcal{W}$ containing x_z^1. Let $\mathcal{W}_C^*$ be the witness structure obtained from $\mathcal{W}_C$ by removing every witness set which intersects W and adding W. Since W contains $W(x_{z_0}^1)$, it is adjacent with every other witness set in $\mathcal{W}_C$ and hence in $\mathcal{W}_C^*$. This implies that any two witness sets in $\mathcal{W}_C^*$ are adjacent to

each other. Hence, if G^* is obtained from G by deleting x, then G^*/F^* is a split graph. Since we are adding at most two edges only for deleted edges in F, we have $|F^*| \leq 2 \cdot |F|$.

> At this stage we rename $G^\star$ to G and $F^\star$ to F.

Condition 5(b): Consider that $W(t) \cap (Y \setminus Y')$ is non-empty. In this case, construction of G^* and F^* is similar to that of Condition 3. Let y be a vertex in $W(t) \cap (Y \setminus Y')$ and S_t be the set of vertices in $W(t) \cap S$ which are adjacent with y via edges in F and Q_t be the set of vertices in $W(t) \cap X$ which are adjacent with y via edges in F. Arbitrarily fix a vertex s_0 in S_t. We add some edges to ensure connectivity of vertices in S_t and some other to ensure adjacency among witness sets in $\mathcal{W}_C$. If $S_t \setminus \{s_o\}$ is an empty set then no edge needs to be included to ensure connectivity. For every vertex s in $S_t \setminus \{s_o\}$, the set $\{s_o, s\}$ is considered by Marking Scheme 4.2. (For any $\alpha > 1$, d is at least two.) For every s in $S_t \setminus \{s_o\}$, vertex y is contained in $M_{CY}(\{s_o, s\})$, the set of common neighbors of $\{s_o, s\}$ in Y. Since y is in $Y \setminus Y'$, and hence unmarked, there is a vertex in $M_{CY}(\{s_o, s\})$ different from y which has been marked for each set $\{s_o, s\}$. For every vertex s in $S_t \setminus \{s_o\}$, let y_s be the marked vertex which is adjacent with s_o and s. We construct F^* from F by following operation: Remove the edge $s_o y$ and for every vertex s in $S_t \setminus \{s_o\}$, remove edge sy in F and add two edges $sy_s, s_o y_s$. Since $W(t)$ also intersects X, there exists another witness set $W(t^\#)$, in $\mathcal{W}_C$ such there is an edge across $W(t^\#)$ and $W(t) \setminus \{y\}$. Let uv be that edge. Include uv in the solution.

Let W be the superset of $W(t) \setminus \{y\}$ which contains the witness sets of all the newly added vertices in $W(t)$. Formally, $W = (W(t) \setminus \{y\}) \cup (\bigcup_{s \in S_t} W(y_s))$ where $W(y_s)$ is the witness set containing y_s. Note that every vertex in S_t is connected with each other (as we have added edges $y_s s, y_s s_o$ for every s in $S_t \setminus \{s_o\}$) and W is adjacent with $W(t^\#)$. Let $\mathcal{W}_C^*$ be the witness structure obtained from $\mathcal{W}_C$ by removing every witness set which intersects with $W, W(t^\#)$ and adding $W \cup W(t^\#)$. Since $W \cup W(t^\#)$ contains $W(t^\#)$, which is adjacent with every other witness set in $\mathcal{W}_C$ and hence in $\mathcal{W}_C'$. This implies that any two witness sets in $\mathcal{W}_C^*$ are adjacent with each other. Hence, if G^* is obtained from G by deleting y then G^*/F^* is a split graph. Since we are adding at most two edges of the form $y_s s, y_s s_o$ for every edge of the form sy in F where $y \in S_t \setminus \{s_o\}$ and at most two edges for edge ys_o, we have $|F^*| \leq 2 \cdot |F|$.

We now argue that even after repeating the process, the size of $|F^*|$ is bounded. In every condition, an edge uv is replaced only if one of the endpoints belongs to $(X \setminus X') \cup (Y \setminus Y')$. Every time edges were replaced by those edges that belong to G'. So once any of the conditions mentioned above consider an edge in F, it will never be considered by any other condition. This implies that the number of edges in F^* is always upper bounded by $2 \cdot |F^\circ|$ where F° is a solution for original instance (G, k). If there exists no witness set **that satisfies any condition**, then $G' = G^*$ and solution F^* is the solution with desired properties. This concludes the proof of the lemma. □

Lemma 4.3 and 4.4 are not sufficient to prove a time-efficient $(2 + \epsilon)$-approximate polynomial kernel. This is primarily because of the additive factor in Lemma 4.3. We present the following lemma, which describes a solution lifting algorithm whose running time is dependent on ϵ.

LEMMA 4.5. *For a fixed α', α satisfying $\alpha' > \alpha > 1$, there exists a solution lifting algorithm which satisfies following properties.*

(1) *Given a reduced instance (G', k) obtained by applying Reduction Rule 4.2 on instance (G, k), partition (S, X, Y) of $V(G)$ and an integer $d = \lceil \frac{1}{\alpha - 1} \rceil$ together with a solution F' for (G', k), it returns a solution F for (G, k) such that $\frac{SpC(G, k, F)}{OPT(G, k)} \leq 2 \cdot \alpha' \cdot \frac{SpC(G', k, F')}{OPT(G', k)}$.*

(2) *The algorithm runs in time $O(m^{\alpha \cdot c + 2})$ where m is the number of edges in G and $c = \frac{1}{\alpha' - \alpha}$.*

PROOF. We present a solution lifting algorithm which considers three cases depending on cardinality of F'. Before mentioning the algorithm, we recall Lemma 4.4 which states that if $OPT(G, k) \leq k$

then $\text{OPT}(G', k) \leq 2 \cdot \text{OPT}(G, k)$. If $\text{OPT}(G, k) = k + 1$ then $\text{OPT}(G', k) \leq k + 1 = \text{OPT}(G, k)$. Hence in either case, $\text{OPT}(G', k) \leq 2 \cdot \text{OPT}(G, k)$.

If cardinality of F' is greater than or equal to $k + 1$ then solution lifting algorithm returns a spanning tree F of G (a trivial solution) as solution for (G, k). In this case, $\text{SpC}(G, k, F) = k + 1 = \text{SpC}(G', k, F')$. Since $\text{OPT}(G', k) \leq 2 \cdot \text{OPT}(G, k)$, in this case we get $\frac{\text{SpC}(G, k, F)}{\text{OPT}(G, k)} \leq 2 \cdot \frac{\text{SpC}(G', k, F')}{\text{OPT}(G', k)}$.

If cardinality of F' is at most k but greater than or equal to c then the algorithm uses Lemma 4.3 to compute a solution F for (G, k) such that cardinality of F is at most $\alpha \cdot |F'| + 1$. Since $\text{OPT}(G', k) \leq 2 \cdot \text{OPT}(G, k)$, this implies:

$$\frac{|F|}{\text{OPT}(G, k)} \leq \frac{2 \cdot (\alpha \cdot |F'| + 1)}{\text{OPT}(G', k)} \leq 2 \cdot \left(\alpha + \frac{1}{|F'|} \right) \cdot \frac{|F'|}{\text{OPT}(G', k)} \leq 2 \cdot \alpha' \cdot \frac{|F'|}{\text{OPT}(G', k)}$$

Last inequality follows from the fact that $|F'| \geq c$ and $1/c = \alpha' - \alpha$. Hence in this case, $\frac{\text{SpC}(G, k, F)}{\text{OPT}(G, k)} \leq 2 \cdot \alpha' \cdot \frac{\text{SpC}(G', k, F')}{\text{OPT}(G', k)}$.

Consider the remaining case when $|F'| < c$. By definition, F' is a valid solution for (G', k). By Lemma 4.3, there exists a solution F for (G, k) such that $|F| \leq \alpha |F'| + 1 < \alpha \cdot c + 1$. Since we are working with minimization problem, this implies $\text{OPT}(G, k) \leq \alpha \cdot c$. In this case, solution lifting algorithm computes an optimum solution for (G, k) by brute force, i.e. checking all set of edges in G of size at most $\alpha \cdot c$, and returns it. In this case, we have, $\frac{\text{SpC}(G, k, F)}{\text{OPT}(G, k)} = 1$.

Hence, the solution lifting algorithm returns a solution with the desired property. The running time of the algorithm follows from its description, and the fact that solution satisfying Lemma 4.3 can be obtained in polynomial time. □

We note that solution lifting algorithm mentioned in Lemma 4.5 allows us to choose a value of α between 1 and α'. This choice of α is a trade-off between the running time of the algorithm (as c is inversely proportional to $\alpha' - \alpha$) and the size of lossy kernel (as d is inversely proportional to $\alpha - 1$). We now present the main result of this section.

THEOREM 1.2. *For any $\epsilon > 0$, SPLIT CONTRACTION parameterized by the size of solution k, admits $(2 + \epsilon)$-approximate polynomial kernelization algorithm which runs in $O(m^{\alpha \cdot c + 2})$ time and return an instance with $O(k^{d^2 + d + 1})$ vertices. Here, m is number of edges in G and constants α, c, and d depend only on ϵ.*

PROOF. Any split graph can have at most one component which contains an edge. Hence, if graph G in given instance (G, k) is not connected, then all but one connected component needs to be contracted to an isolated vertex. If there are more than one connected components with at least $k + 1$ vertices, then we return a trivial instance. Otherwise, we apply reduction rules to the unique connected component in G, which has more than $k + 1$ vertices. Hence, we can assume that the input graph is connected.

For a given instance (G, k), a kernelization algorithm applies Reduction Rule 4.1. If it returns a trivial instance then the statement is vacuously true. If it does not return a trivial instance then the algorithm partitions $V(G)$ in three sets (S, X, Y) such that $|S| \leq 10k$ and $G - S$ is a split graph with (X, Y) as its split partition. For given ϵ, fix $\alpha' = 1 + \epsilon/2$. The algorithm fixes α which is strictly more than 1 and strictly less than α'. The algorithm then applies Reduction Rule 4.2 on (G, k) with partition (S, X, Y) and α. The algorithm returns the reduced instance as an $(2 \cdot \alpha')$-lossy kernel for (G, k).

The correctness of the algorithms follows from Lemma 4.4 and 4.5 combined with the fact that Reduction Rule 4.2 is applied at most once. By Observation 2.2, Reduction Rule 4.1 can be applied in polynomial time. The size of the instance returned by Reduction Rule 4.2 is at most

$O(k^{d^2+d+1})$. Reduction Rule 4.2 can be applied in time $n^{O(1)}$ if number of the vertices in G is more than $O(k^{d^2+d+1})$.   □

5   LOWER BOUND ON INAPPROXIMABILITY OF SPLIT CONTRACTION

In this section, we show that for any $\delta > 0$, SPLIT CONTRACTION parameterized by the solution size does not admit a factor $(5/4 - \delta)$-FPT approximation algorithm, assuming Gap-ETH. Towards this, we give a polynomial time reduction from MULTICOLORED DENSEST-k-SUBGRAPH WITH PERFECT COMPLETENESS, which is $k^{o(1)}$-factor FPT inapproximable (Corollary 5.1). In the DENSEST-k-SUBGRAPH problem, input is a graph G and an integer k; and the goal is to find a subset S of $V(G)$ of size k such that $\mathrm{Den}(S) \geq k^{-g(k)}$. Recall that we defined $\mathrm{Den}(S) = |E(G[S])|/\binom{|S|}{2}$. DENSEST-$k$-SUBGRAPH WITH PERFECT COMPLETENESS is a special case of DENSEST-k-SUBGRAPH problem, in which the input graph G contains a k-clique, and it is known to be $k^{o(1)}$-factor FPT inapproximable [14]. Now, we define a colorful version of DENSEST-k-SUBGRAPH WITH PERFECT COMPLETENESS. We call a set of edges *colorful*, if all the edges in the set are colored with pairwise distinct colors. We say that a clique is *colorful-clique*, if the set of edges in this clique is colorful. In the MULTICOLORED DENSEST-k-SUBGRAPH WITH PERFECT COMPLETENESS, an input graph G is given with an edge coloring $\phi : E(G) \to [\binom{k}{2}]$, and G is promised to have a colorful-k-clique. The goal is to find a subset S of $V(G)$ of size k such that the set $E(G[S])$ is colorful and $\mathrm{Den}(S) \geq k^{-g(k)}$.

We start with the following known inapproximability result for DENSEST-k-SUBGRAPH WITH PERFECT COMPLETENESS.

PROPOSITION 5.1 (LEMMA 5.21 IN [14]).   *Assuming* Gap-ETH, *for every function $g = o(1)$ and every function f, there is no $f(k) \cdot n^{O(1)}$-time algorithm such that, given an integer k and a graph G with n vertices containing a k-clique, always outputs a set S of k vertices such that $\mathrm{Den}(S) \geq k^{-g(k)}$.*

For notational convenience, let $t = \binom{k}{2}$. The size of $(|E(G)|, t)$-perfect hash family is bounded by $e^t t^{O(\log t)} \cdot n^{O(1)}$ [42]. Using $(|E(G)|, t)$-perfect hash family and Proposition 5.1, we obtain the following result for MULTICOLORED DENSEST-k-SUBGRAPH WITH PERFECT COMPLETENESS.

COROLLARY 5.1.   *Assuming* Gap-ETH, *for every function $g = o(1)$ and every function f, there is no $f(k) \cdot n^{O(1)}$-time algorithm such that, given an integer k and an edge colored graph G with n vertices containing a colorful-k-clique, always outputs a set S of k vertices such that $E(G[S])$ is colorful and $\mathrm{Den}(S) \geq k^{-g(k)}$.*

For a given subset S of vertices, we say that S *spans* an edge, if both of its endpoints are in S. Due to Corollary 5.1, we obtain following result.

COROLLARY 5.2.   *Assuming* Gap-ETH, *for every $0 < \epsilon < 1$, and for every function f, there is no $f(k) \cdot n^{O(1)}$-time algorithm such that, given an integer k and an edge colored graph G with n vertices containing a colorful-k-clique, always outputs a set S of k vertices which span at least $\epsilon \binom{k}{2}$ colorful edges.*

In the following lemma, we strengthen the above result.

LEMMA 5.1.   *Assuming* Gap-ETH, *for every $0 < \epsilon < 1$, $\alpha > 1$ and every function f, there is no $f(k) \cdot n^{O(1)}$-time algorithm such that, given an integer k and an edge colored graph G with n vertices containing a colorful-k-clique, always outputs a set S of at most αk vertices that spans at least $\epsilon \binom{k}{2}$ colorful edges.*

PROOF.   Assume that there is a $f(k) \cdot n^{O(1)}$-time algorithm, $\mathcal{A}$, that takes the input (G, k) such that G has a colorful-k-clique, and outputs a set S of at most αk vertices, that spans at least $\epsilon \binom{k}{2}$

colorful edges. Partition the set S into $\lceil 2\alpha \rceil$ sets $S_1, S_2, \cdots, S_{\lceil 2\alpha \rceil}$ each of size at most $k/2$. As the set S spans at least $\epsilon\binom{k}{2}$ colorful edges, there exists a pair of sets, say S_i, S_j, in this partition, such that $S_i \cup S_j$ spans at least $\epsilon'\binom{k}{2}$ colorful edges, where $\epsilon' = \epsilon/\binom{\lceil 2\alpha \rceil}{2}$. Moreover, the size of $S_i \cup S_j$ is at most k. This implies that there exists an algorithm, that outputs a set of vertices of size at most k that spans $\epsilon'\binom{k}{2}$ colorful edges, in time $f(k) \cdot n^{O(1)}$. Hence, the existence of such algorithm contradicts Corollary 5.2. $\square$

Now, we are ready to give our reduction. We argue that if SPLIT CONTRACTION parameterized by the solution size admits a factor $(5/4 - \delta)$-FPT approximation algorithm for some δ, then it contradicts Lemma 5.1, and hence Gap-ETH. Towards this, we present a reduction in which given an instance (G, k) of MULTICOLORED DENSEST-k-SUBGRAPH WITH PERFECT COMPLETENESS, and a constant δ, constructs an instance (G', k') of SPLIT CONTRACTION.

Reduction Algorithm: Given an instance (G, k) of MULTICOLORED DENSEST-k-SUBGRAPH WITH PERFECT COMPLETENESS, and a constant $\delta > 0$, the algorithm constructs a graph G' as follows. Recall that $t = \binom{k}{2}$.

- Fix $\rho = \lceil \delta t/k \rceil$, $k' = 2t + \rho k$, and $k^\circ = \lceil 5/2 \cdot k' \rceil + 2$.
- For every vertex u in $V(G)$, add ρ copies of it to $V(G')$ and convert them into a clique. Formally, define $X_u := \{u_1, \cdots, u_\rho\}$ for every vertex u in $V(G)$. Let $Z = \cup_{u \in V(G)} X_u$. We add $k^\circ + 2$ extra vertices in Z. For every pair of vertices $z_1, z_2 \in Z$, add an edge z_1z_2 to $E(G')$. That is, the vertices in Z form a clique.
- For every vertex z in Z, we add k° pendant vertices $y_1, \cdots, y_{k^\circ}$ to $V(G')$. We denote the set of these vertices as Guard_V. We also add edges $zy_1, \cdots, zy_{k^\circ}$ to $E(G')$.
- For a given coloring function $\phi : E(G) \rightarrow [t]$, let E_i denote the set of edges, which are assigned color i. For every $i \in [t]$, we create a set of vertices ES_i as follows. We add a vertex w_e corresponding to every edge e in E_i. Formally, $\mathsf{ES}_i = \{w_e \mid e \in E_i\}$. For every $i \in [t]$, we add ES_i to $V(G')$. We call these sets *edge selector sets*. Let $e = uv$ be an edge in G, we add all the edges between w_e and the vertices in $Z \setminus (X_u \cup X_v)$ to $E(G')$. Let $\mathsf{ES} = \cup_{i \in [t]} \mathsf{ES}_i$.
- For every ES_i, we add a vertex g_i to $V(G')$. We call this vertex *cap vertex*, corresponding to ES_i. We also add all the edges between g_i and the vertices in ES_i. We denote the set of cap vertices by Cap.
- For every cap vertex g_i in $V(G')$, we add k° pendant vertices $y_i^1, \cdots, y_i^{k^\circ}$ to $V(G')$. We denote the set of these vertices as Guard_E. We also add edges $g_iy_i^1, \cdots, g_iy_i^{k^\circ}$ to $E(G')$.
- We add a set of t *special vertices*, $\mathsf{SV} = \{s_1, \cdots, s_t\}$ to $V(G')$. We add all the edges between s_i and the vertices in ES_i, where $i \in [t]$. We add edges between every pair of vertices in SV, that is, for all $s_i, s_j \in \mathsf{SV}$, we add an edge s_is_j to $E(G')$. That is, the vertices in SV form a clique. We call s_i as *special vertex* corresponding to ES_i.
- For every special vertex x in SV, we add k° pendant vertices $x_1, \cdots, x_{k^\circ}$ to $V(G')$. We denote the set of these vertices as Guard_S. We also add edges $xx_1, \cdots, xx_{k^\circ}$ to $E(G')$.

This completes the construction. Reduction algorithm returns (G', k') as an instance of SPLIT CONTRACTION. See Figure 5 for an illustration.

Note that without loss of generality, we can assume that for a given instance (G, k) of MULTICOLORED DENSEST-k-SUBGRAPH WITH PERFECT COMPLETENESS, graph G contains at least $k + 1$ vertices, and k is not a constant.

LEMMA 5.2. *Let (G', k') be the instance of SPLIT CONTRACTION returned by the reduction algorithm mentioned above when the input is (G, k) and $\delta > 0$. Then, there exists a set of edges F' in G' of size most k', such that G'/F' is a split graph.*
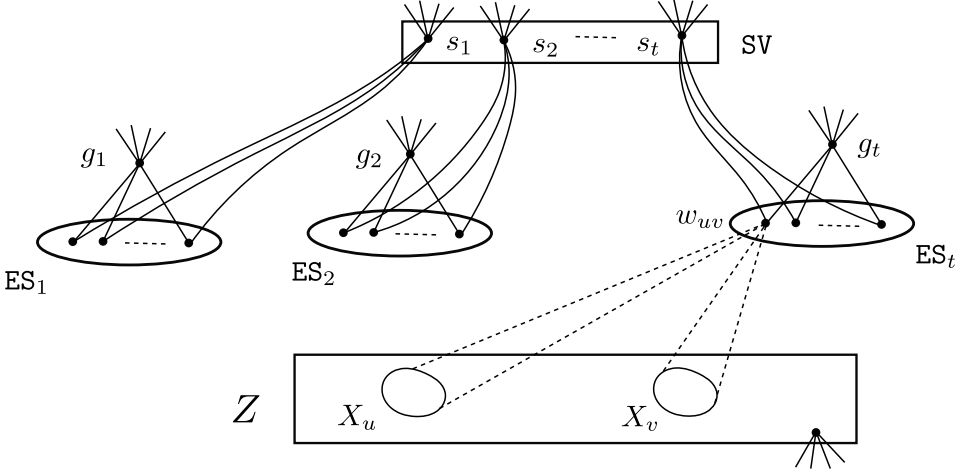
Fig. 5. Sets with rectangular boundary represent cliques whereas sets with elliptical boundary represent independent sets. A vertex w_{uv}, corresponding to an edge uv, is adjacent to all vertices in Z except the ones in $X_u \cup X_v$. Dashed lines shows non-adjacency between the vertex and sets. Please refer to reduction from MULTICOLORED DENSEST-k-SUBGRAPH WITH PERFECT COMPLETENESS to SPLIT CONTRACTION for details.

PROOF. Let $S = \{u^1, \cdots, u^k\}$ be a set of vertices that induces a colorful-k-clique in (G, k). Let $E_S = \{e_1, \cdots, e_t\}$ be the set of edges in $G[S]$. Since E_S is a set of colorful edges, let $e_i \in E_i$, where i in $[t]$. Let $W = \{w_{e_i} \in V(G') \mid e_i \in E_S\}$, we construct a solution F' to (G', k') as follows. For every $w_{e_i} \in W$, add the edges $w_{e_i}g_i, w_{e_i}s_i \in E(G')$ to F', where g_i is the cap vertex corresponding to ES_i, and s_i is the special vertex corresponding to ES_i. Note that we have added $2t$ edges to F'. As the number of vertices in $V(G)$ is at least $k + 1$, there exists a vertex, say u_0, in $V(G) \setminus S$. Consider a vertex z_0 in X_{u_0} in graph G'. For every vertex u in S, we add the edges $\{u_1z_0, \cdots, u_\rho z_0\}$ to F'. Thus, for every $u \in S$, we have added ρ edges to F'. Hence, $|F'| = 2t + \rho k = k'$.

We now show that G'/F' is a split graph. Let New be the set of new vertices that are introduced in G'/F' by contracting edges in F'. Let $C = (Z \setminus (\cup_{u \in S} X_u \cup \{z_0\})) \cup \mathsf{New}$. Let $I = \mathsf{Guard}_V \cup \mathsf{Guard}_E \cup \mathsf{Guard}_S \cup (\mathsf{ES} \setminus W)$. We claim that (C, I) is a split partition of G'/F'. By the construction of G', $V(G') = Z \cup \mathsf{ES} \cup \mathsf{Cap} \cup \mathsf{SV} \cup \mathsf{Guard}_V \cup \mathsf{Guard}_E \cup \mathsf{Guard}_S$. Since $\{z_0\} \cup (\cup_{u \in S} X_u) \cup W \cup \mathsf{Cap} \cup \mathsf{SV} \subseteq V(F')$, we know that (C, I) is a partition of G'/F'. As I is an independent set in G' and no edges incident to I are contracted, this set is also an independent set in G'/F'.

We now argue that C is a clique in G'. Since every pair of vertices in Z is adjacent to each other in G'/F', if the vertices u, v are in $Z \setminus \mathsf{New}$, then they are adjacent to each other in G'/F'. Consider a vertex u in New; we have the following two cases.

Case (A): Vertex u is obtained by contracting $w_{e_i}g_i$ and $w_{e_i}s_i$ for some edge $e_i(= xy)$ in E_i. By the construction of G', the vertex w_{e_i} is adjacent to all the vertices in $Z \setminus (X_x \cup X_y)$ of the graph G'. Hence, u is adjacent to all the vertices, which are in $Z \setminus (X_x \cup X_y)$, thus, u is adjacent to all the vertices in $Z \setminus (\cup_{u \in S} X_u \cup \{z_0\})$. We now show that u is also adjacent to all the vertices in New. Let $v \in \mathsf{New}$, and suppose that v is a vertex obtained by contracting $w_{e_j}g_j$ and $w_{e_j}s_j$, where $e_j \in E_S$. Since s_i, s_j are in SV, $s_is_j \in E(G')$, u, v are adjacent to each other in G'/F'. Now, suppose that v is obtained by contracting x_iz_0, where $x \in S$ and $i \in [\rho]$. Since $u_0 \notin S$, z_0 is adjacent to w_{e_i}, uv is an edge in $E(G'/F')$.

Case (B): Vertex u is obtained by contracting the edge $x_i z_0$, where $x \in S$. Since $z_0 \in Z$, u is adjacent to all the vertices in $Z \setminus (\cup_{u \in S} X_u \cup \{z_0\})$. Now, consider another vertex $v \in \mathsf{New}$. Note that as u, v are two different vertices in $V(G'/F')$, v can not be obtained by contracting $y_i z_0$ for any $y \in S$. Thus, v is obtained by contracting $w_{e_j} g_j$ and $w_{e_j} s_j$, where $e_j \in E_S$. Then, as argued above uv is an edge in $E(G'/F')$ (we only need to interchange u and v in the previous argument). Hence, any two vertices in C are adjacent.

This implies that (C, I) is a split partition of G'/F', and hence G'/F' is a split graph. Since the number of edges in F' is at most k', this proves the lemma. □

In the following lemma, we argue that given an approximate solution for an instance of Split Contraction, one can obtain a set of αk vertices, $\alpha > 1$, that spans at least ϵt colorful edges, where $0 < \epsilon < 1$.

LEMMA 5.3. *Let (G', k') be an instance of Split Contraction returned by the reduction algorithm mentioned above when the input is (G, k) and $\delta > 0$. If there exists a set of edges F' in G' such that G'/F' is a split graph and size of F' is at most $(5/4 - \delta)k'$, then there exists a set of at most $1/\delta \cdot k$ vertices in G that spans at least $3\delta/2 \cdot t$ colorful edges.*

We establish some properties of the instance of Split Contraction that is returned by the reduction algorithm and its solution before presenting proof of Lemma 5.3. In Claims 5.1 to 5.7, we use the following notation: The reduction algorithm returns (G', k'), when input is (G, k), and δ. Let F' be a set of edges in G' such that G'/F' is a split graph and the size of F' is at most $(5/4-\delta) \cdot k'$. Let $\mathcal{W}$ be the G'/F'-witness structure of G'. Let $\psi : V(G') \rightarrow V(G'/F')$ be the onto function corresponding to contracting all the edges in F'. For a vertex $\tilde{w}$ in $V(G'/F')$, $W(\tilde{w})$ denotes the witness set which is contracted to obtain the vertex $\tilde{w}$. Hence, for a vertex u in $V(G')$, if $u \in W(\tilde{w})$ then $\psi(u) = \tilde{w}$. We fix a split partition $(\tilde{C}, \tilde{I})$ of $V(G'/F')$. Note that $|F'| + 2 \leq 2|F'| \leq 2(5/4 - \delta)k' \leq 5/2k' \leq k^{\circ}$. Hence, there are at least $|F'| + 2$ many pendant vertices adjacent to every cap vertices, special vertices, and every vertex in Z of G'. Moreover, since the size of Z is at least $k^{\circ} + 2$, there exists at least one vertex in Z which is not in $V(F')$. Let $z^{\star}$ be one such vertex in $V(G')$ which is in $Z \setminus V(F')$. Note that $W(\psi(z^{\star})) = \{z^{\star}\}$.

We refer readers to Section 1 for an overview of the proof. Consider a cap vertex g_i and the witness set $W(\psi(g_i))$. A cap vertex g_i is said to be *spoiled*, if $W(\psi(g_i))$ either (a) contains another cap vertex; (b) intersects with Z; or (c) has more than one vertex from ES. We bound the number of cap vertices that can be spoiled because of (a), (b) or (c) in Claims 5.5, 5.6, and 5.7, respectively. We first present few results, which are used in the proof of these claims.

CLAIM 5.1. *If $u \in V(G')$ is either a cap vertex or a special vertex or in Z, then $\psi(u)$ is in $\tilde{C}$.*

PROOF. Any such vertex u is adjacent to $|F'| + 2$ many pendant vertices. This implies that there are at least two pendant vertices, say u_1, u_2, which are not in $V(F')$, which in turn implies that $W(\psi(u_1))$ and $W(\psi(u_2))$ are singleton sets in $\mathcal{W}$. Since u_1, u_2 are not adjacent to each other in G' and $W(\psi(u_1)), W(\psi(u_2))$ are singleton witness sets, $\psi(u_1), \psi(u_2)$ are not adjacent to each other in G'/F'. Hence, at most one of them can be in $\tilde{C}$. Without loss of generality, let $\psi(u_1)$ is in $\tilde{I}$. Since u is adjacent to u_1 in G', and u is not contained in $\psi(u_1)$, $\psi(u)$ is adjacent to $\psi(u_1)$. This implies $\psi(u)$ is in $\tilde{C}$. □

CLAIM 5.2. *For every cap vertex g_i there exists a vertex u_e in ES_i such that $g_i u_e$ is in F.*

PROOF. Recall that $\psi(z^{\star})$ is a vertex in $\tilde{C}$ and $W(\psi(z^{\star}))$ is a singleton witness set. Assume that for a cap vertex g_i, there is no vertex u_e in ES_i such that edge $g_i u_e$ is in F. This implies that $\psi(g_i) \cap \mathsf{ES}_i$ is an empty set. Since neighbors of g_i outside ES_i (i.e. pendant neighbors of g_i) are not adjacent

to $z^\star$ in G', there is no edge with one endpoint in $W(\psi(g_i))$ and another one in $W(\phi(z^\star))$. But by Claim 5.1, both $\psi(z^\star)$ and $\psi(g_i)$ are in $\tilde{C}$. This contradicts the fact that $\tilde{C}$ is a clique. Hence, our assumption is wrong, which concludes the proof of the claim. □

CLAIM 5.3. *For every special vertex s_i there exists a vertex u_e in ES such that u_e is in $W(\psi(s_i))$.*

PROOF. For the sake of contradiction, assume that there exists a witness set, say $W(\psi(s_i))$ such that $W(\psi(s_i)) \cap \mathsf{ES} = \emptyset$. Recall that $\psi(z^\star)$ is a vertex in $\tilde{C}$ and $W(\psi(z^\star))$ is a singleton witness set. Since $W(\psi(s_i))$ does not contain any vertex of ES, $W(\psi(s_i)) \subseteq \mathsf{SV} \cup \mathsf{Guard}_S$. Hence, there is no edge with one endpoint in $W(\psi(s_i))$ and another one in $W(\psi(z^\star))$. By Claim 5.1 both $\psi(s_i)$ and $\psi(z^\star)$ are in $\tilde{C}$. This contradicts the fact that $\tilde{C}$ is a clique. Hence, our assumption is wrong, which concludes the proof of the claim. □

CLAIM 5.4. *There are at least $2t$ edges in F', which are incident to either cap vertices, or special vertices. Moreover, every vertex can be assigned to an edge in F', which is unique to it.*

PROOF. By Claim 5.2, there are at least t edges incident to cap vertices. Since a cap vertex is neither adjacent to other cap vertex nor to a special vertex, a cap vertex can be uniquely mapped to the edge incident to it.

Consider a special vertex s_i. Define $S = W(\psi(s_i)) \cap \mathsf{SV}$. By Claim 5.3, $W(\psi(s_i))$ contains a vertex, say u_e, which is in ES. Hence, there is an additional edge, which is incident to S. Without loss of generality, we can assume that F' contains a spanning tree of $G'[S \cup \{u_e\}]$. This implies that there are at least $|S|$ many edges incident to S. To assign each vertex to a unique edge, we root this spanning tree at u_e. For every vertex in S, assign it to the edge connecting that vertex to its parent in this rooted tree. Since witness structure $\mathcal{W}$, partitions SV, there are at least $|\mathsf{SV}| = t$ edges in F' which has at least one endpoint in SV. Note that none of these edges is incident to cap vertices. This concludes the proof. □

Claim 5.4 allows us to define a *one-to-one* function λ from the set of cap vertices and special vertices to the edges in F' such that $\lambda(v)$ is an edge incident to v. We call such functions as *accounting functions*. We use such a function in the following arguments to bound certain kinds of witness sets. For an accounting function λ, the set of edges in F' which do not belong to the range of λ are called extra edges. We fix a function λ and modify it in Claim 5.5 and 5.7 at certain vertices of special vertices to obtain another accounting function. We use the number of edges in extra edges with respect to this function to argue that the number of spoiled cap vertices is not very large. Let $\gamma = 3\delta/4$. Note that, for any accounting function λ, the number of edges in extra edges is at most $|F'| - 2t \le (5/4 - \delta)k' - 2t \le (5/4 - \delta) \cdot (2t + \rho k) - 2t \le (1/2)t - (3\delta/4)t = (1/2 - \gamma) \cdot t$. If an edge is not incident to cap or special vertex then such an edge is always in extra edges for any accounting function.

CLAIM 5.5. *Let Cap_1 be a subset of Cap such that every vertex in Cap_1 is in a witness set in $\mathcal{W}$ that contains at least two cap vertices. Then, F' contains at least $1/2 \cdot |\mathsf{Cap}_1|$ edges in extra edges for an accounting function λ.*

PROOF. Let $\mathcal{W}_1$ be the collection of witness sets in $\mathcal{W}$ which contain at least two cap vertices. Clearly, $|\mathcal{W}_1| \le 1/2 \cdot |\mathsf{Cap}_1|$. Consider a witness set $W(p)$ in $\mathcal{W}_1$, we argue that there are at least $|W(p) \cap \mathsf{Cap}_1| - 1$ many edges in extra edges incident to the vertices of $W(p)$ for an accounting function λ.

Consider the accounting function λ which is constructed/modified before considering $W(p)$. We modify this accounting function for some vertices in SV to obtain another accounting function. Let T be a spanning tree of $G'[W(p)]$ such that $E(T) \subseteq F'$. We arbitrarily fix a cap vertex g_o in $W(p)$

and the root of T at g_o. Consider a cap vertex g_i and let g_j be the first cap vertex on the unique path from g_i to g_o in T(Note that g_j can be equal to g_o). Let $P_{g_i g_j}$ be the unique path between g_i and g_j in the tree T. By the construction of G', there exists a vertex u_e^i in ES_i such that $g_i u_e^i$ is an edge in $E(P_{g_i g_j})$. We modify λ in a way that there exists an edge incident to u_e^i which is in extra edges. Since u_e^i is not in the domain of accounting function, this edge in extra edges is unique to g_i. Hence, for every cap vertex in $W(p) \cap \mathsf{Cap}_1$ except for g_o, there is an edge in extra edges with respect to an accounting function λ.

Consider the path $P_{g_i g_j}$ between g_i and g_j in the spanning tree T. Clearly, there exists a vertex $u_e^i \in \mathsf{ES}_i$ and a vertex $w \in Z \cup \{s_i\}$ such that $g_i u_e^i, u_e^i w \in E(P_{g_i g_j})$, otherwise there cannot be a path between g_i and g_j. If $w \in Z$, then $u_e^i w$ is in extra edges for λ. Suppose that $w = s_i$. Then, in the path $P_{g_i g_j}$, either s_i is adjacent to $u_{e'}^i \in \mathsf{ES}_i$ or a special vertex $s_q \in \mathsf{SV}$, then $s_i u_{e'}^i \in E(P_{g_i g_j})$. Note that either $s_i u_e^i$ or $s_i u_{e'}^i$ is in extra edges for λ. Now, suppose that $s_i s_q \in E(P_{g_i g_j})$. If $\lambda(s_i) = s_i s_q$, then $s_i u_e^i$ is in extra edges for λ. Suppose that $\lambda(s_i) = s_i u_e^i$. Now, we modify λ to obtain $\lambda(s_i) = s_i s_q$. We denote the successor and predecessor of a vertex s in the path $P_{g_i g_j}$ by $\mathsf{succ}[s]$ and $\mathsf{pred}[s]$, respectively. Let s_ℓ be the first special vertex in the path $P_{g_i g_j}$ such that either $\lambda(s_\ell) = s_\ell \mathsf{succ}[s_\ell]$ or $\mathsf{succ}[s_\ell] \in \mathsf{ES}_\ell$. Let $\widetilde{\mathsf{SV}}$ be the set of the vertices in the subpath of $P_{g_i g_j}$ from s_i to s_ℓ. For every special vertex s in $\widetilde{\mathsf{SV}}$, we set $\lambda(s) = s\mathsf{succ}[s]$. Note that λ is still a *one-to-one* function as earlier either $\lambda(s_\ell)$ (when $\lambda(s_\ell) = s_\ell \mathsf{succ}[s_\ell]$) or $\lambda(s_\ell)$ (when $\lambda(s_\ell) = s_\ell \mathsf{pred}[s_\ell]$, and $\mathsf{succ}[s_\ell] \in \mathsf{ES}_\ell$) was not in the range of λ. Now, since $\lambda(s_i) = s_i s_q$, $s_i u_e^i$ is in extra edges for the modified λ. This implies that for any cap vertex $g_i \neq g_0$ there is an edge incident to u_e^i which is in extra edges for an accounting function λ. Hence, F' has at least $|W(p) \cap \mathsf{Cap}_1| - 1$ edges in extra edges.

By the above discussion, for every witness set $W(p) \in \mathcal{W}_1$, F' has at least $|W(p) \cap \mathsf{Cap}_1| - 1$ edges in extra edges for the function λ. Since $\mathcal{W}$ partitions vertices in Cap, we can infer that F' has at least $|\mathsf{Cap}_1| - |\mathcal{W}_1|$ edges in extra edges for the function λ. □

CLAIM 5.6. *Let Cap_2 be a subset of Cap such that every vertex in Cap_2 is in a witness set in $\mathcal{W}$ that intersects with Z and contains exactly one cap vertex. Then, F' contains at least $|\mathsf{Cap}_2|$ edges in* extra edges *for any function λ.*

PROOF. Let $\mathcal{W}_2$ be the collection of witness sets in $\mathcal{W}$ that intersects with Z and contains exactly one cap vertex. Consider a witness set $W(p)$ in $\mathcal{W}_2$ and let z be a vertex in $W(p) \cap Z$. Hence, F' contains at least one edge incident to z. For any function λ, an edge incident to a vertex of Z is in extra edges as the vertices of Z are neither adjacent to cap vertices nor special vertices. This implies that F' contains an edge in extra edges that is incident to a vertex of $W(p)$. Since $\mathcal{W}$ partitions vertices in Cap_2, and every witness set in $\mathcal{W}_2$ contains exactly one cap vertex, F' has at least $|\mathsf{Cap}_2|$ edges in extra edges for the function λ. □

CLAIM 5.7. *Let Cap_3 be a subset of Cap such that every vertex in Cap_3 is in a witness set in $\mathcal{W}$ that is disjoint from Z; contains exactly one cap vertex; and at least two vertices of ES. Then, F' contains at least $|\mathsf{Cap}_3|$ edges in* extra edges *for a function λ.*

PROOF. Let $\mathcal{W}_3$ be a collection of witness sets in $\mathcal{W}$ that are disjoint from Z; contains exactly one cap vertex; and at least two vertices of ES. Consider a witness set $W(p)$ in $\mathcal{W}_3$. We argue that F' has at least one edge in extra edges that is incident to a vertex of $W(p)$ for a function λ.

Let T be a spanning tree of $G'[W(p)]$ such that $E(T) \subseteq F'$. Consider the accounting function λ which is constructed/modified before considering $W(p)$. Let g_i be the cap vertex contained in $W(p)$. By Claim 5.2, there exists $u_e^i \in \mathsf{ES}_i$ such that $g_i u_e^i \in F'$. We consider two cases depending on whether another vertex in $\mathsf{ES} \cap W(p)$ is in ES_i or not. Suppose that there exists a vertex $u_{e'}^i \in \mathsf{ES}_i \cap W(p)$ such that $u_{e'}^i \neq u_e^i$ and if $g_i u_{e'}^i \in E(T)$, then either $g_i u_e^i$ or $g_i u_{e'}^i$ is in extra edges for λ. Suppose

that $g_i u^i_{e'} \notin E(T)$ and consider a path from g_i to $u^i_{e'}$, say $P_{g_i u^i_{e'}}$, in the spanning tree T. Since $W(p)$ does not intersect with Z, $E(P_{g_i u^i_{e'}}) = \{g_i u^i_e, u^i_e s_i, s_i u^i_{e'}\}$, either $s_i u^i_e$ or $s_i u^i_{e'}$ is in `extra edges` for λ. Now, suppose that u^i_e is the only vertex in $\mathsf{ES}_i \cap W(p)$. Since $W(p)$ contains at least two vertices of ES, there exists a vertex $u^j_{e'} \in \mathsf{ES}_j \cap W(p)$, where $j \in [t], j \neq i$. Consider a path $P_{g_i u^j_{e'}}$ from g_i to $u^j_{e'}$, since $W(p)$ is disjoint from Z, $u^i_e s_i, u^j_{e'} s_j \in E(P_{g_i u^j_{e'}})$ (otherwise there can not be a path from g_i to $u^j_{e'}$), and the path from s_i to s_j in T contains only special vertices. Let S be the set of vertices in the path from s_i to s_j in T. We modify λ to define $\lambda(s) = s\mathsf{succ}[s]$, for all $s \in S$. Note that λ is still a *one-to-one* function, and $s_i u^i_e$ is in `extra edges`. Hence, for every witness set in Cap_3, we have an edge in `extra edges` for some function λ.   □

PROOF. (of Lemma 5.3) Let $\mathsf{Cap}_1, \mathsf{Cap}_2$, and Cap_3 be the subset of Cap as defined in Claim 5.5, 5.6 and 5.7. Note that sets $\mathsf{Cap}_1, \mathsf{Cap}_2, \mathsf{Cap}_3$ are pairwise disjoint. Let Cap_4 be the collection of cap vertices in $\mathsf{Cap} \setminus (\mathsf{Cap}_1 \cup \mathsf{Cap}_2 \cup \mathsf{Cap}_3)$.

We first argue that $|\mathsf{Cap}_4| \geq 2\gamma t$. Since there are t many cap vertices, $|\mathsf{Cap}_1| + |\mathsf{Cap}_2| + |\mathsf{Cap}_3| + |\mathsf{Cap}_4| = t$. Recall the accounting function λ, which was fixed before Claim 5.5. We modify this function in Claim 5.5 and 5.7 at some special vertices to obtain another accounting function. Note that the modifications to λ in Claim 5.5 are at special vertices, which are contained in witness sets that contain at least two cap vertices. The modifications to λ in Claim 5.7 are at special vertices, which are contained in witness sets that contain exactly one cap vertex. Since witness sets partition special vertices, one modification does not affect another. Since Claim 5.6 holds true for any accounting function, we know that for the function λ, there are at least $1/2 \cdot |\mathsf{Cap}_1| + |\mathsf{Cap}_2| + |\mathsf{Cap}_3|$ many edges in `extra edges`. Since there are at most $(1/2 - \gamma)t$ many edges in `extra edges` for any accounting function, $|\mathsf{Cap}| - |\mathsf{Cap}_4| = |\mathsf{Cap}_1| + |\mathsf{Cap}_2| + |\mathsf{Cap}_3| \leq (1 - 2\gamma)t$. Hence, $|\mathsf{Cap}_4| \geq 2\gamma t$. This implies that there are at least $2\gamma t$ many cap vertices, which are contained in a witness set, that does not contain any other cap vertex, no vertex of Z, and precisely one vertex of ES.

Let $\mathcal{W}^\star$ be the subset of witness sets that contain at least one cap vertex. Let $\mathcal{W}_1, \mathcal{W}_2$, and $\mathcal{W}_3$ be the subset of $\mathcal{W}^\star$ as defined in the proofs of Claim 5.5, 5.6 and 5.7. Let $\mathcal{W}_4$ be the collection of remaining witness sets in $\mathcal{W}^\star \setminus (\mathcal{W}_1 \cup \mathcal{W}_2 \cup \mathcal{W}_3)$. Note that any witness set in $\mathcal{W}_4$ contains exactly one cap vertex, no vertex of Z, and exactly one vertex of ES. Hence, $|\mathcal{W}_4| = |\mathsf{Cap}_4| \geq 2\gamma t$.

Let $W(p)$ be a witness set in $\mathcal{W}_4$ and u_e be a vertex in $W(p) \cap \mathsf{ES}$. We argue that all the non-neighbors of u_e in Z are in $V(F')$. For the contradiction, suppose that there exists a non-neighbor of u_e, say w, in Z that does not belong to $V(F')$. Note that $W(p)$ contains a cap vertex. By Claim 5.1, $\psi(w)$ and p are in $\tilde{C}$. Since $W(p)$ neither contains a vertex of Z nor any other vertex of ES, p and $\psi(w)$ are not adjacent to each other, a contradiction to the fact that $\tilde{C}$ is a clique. Hence, our assumption is wrong and all the non-neighbors of u_e in Z are in $V(F')$. Recall that for every vertex v in G, the reduction algorithm has added ρ many copies of v in G', and the set of these vertices is denoted by X_v. If the vertex u_e in graph G' corresponds to the edge $e = v_1 v_2$ then all vertices in $X_{v_1} \cup X_{v_2}$ are contained in $V(F')$.

Let Y be the subset of ES such that every vertex in Y is in a witness set in $\mathcal{W}_4$. Since each witness set in $\mathcal{W}_4$ contains exactly one vertex of Y, $|Y| \geq 2\gamma t$. Let Y_E be the set of edges in G, which corresponds to vertices in Y. Let $\bar{N}_Z(Y)$ be the set of non-neighbors of Y in Z. We know that $\bar{N}_Z(Y) \subseteq V(F')$. Since the size of F' is at most $|F'| \leq (5/2)t - \gamma t$ and $2t$ edges are incident to $\mathsf{Cap} \cup \mathsf{SV}$ (Claim 5.4) there are at most $t/2 - \gamma t$ edges incident to the vertices of Z. Since every edge can be incident to at most two vertices of $\bar{N}_Z(Y)$, $|\bar{N}_Z(Y)| \leq t - 2\gamma t$. Let S be the set of vertices, which are endpoint of some edge in Y_E in G. Since we have added ρ copies for each vertex in S, we have $\rho|S| = |\bar{N}_Z(Y)| \leq t - 2\gamma t$. This implies there are at most $(1 - 2\gamma)t/\rho \leq k/\delta$ (as $\rho = \lceil \delta t/k \rceil$) vertices which span at least $2\gamma t$ many edges in G. Since a witness set in $\mathcal{W}_4$ contains exactly one cap vertex

and exactly one vertex of ES, due to Claim 5.2, any two edges in Y_E correspond to two vertices in G', which are in different edge selector sets, that correspond to different color class in edge coloring ϕ in G. Hence, the set of edges Y_E is colorful. This completes the proof of the lemma. □

We are now in a position to prove the main theorem of this section.

THEOREM 1.3. *Assuming* Gap-ETH, *no FPT time algorithm can approximate* SPLIT CONTRACTION *within a factor of* $(5/4 - \delta)$, *for any fixed constant* $\delta > 0$.

PROOF. For the sake of contradiction, assume that for a given fixed $\delta > 0$, there exists an FPT time algorithm, say $\mathcal{A}_\delta$, which can approximate SPLIT CONTRACTION within a factor of $(5/4 - \delta)$.

Consider an instance (G, k) of MULTICOLORED DENSEST-k-SUBGRAPH WITH PERFECT COMPLETE-NESS. We run the reduction algorithm mentioned in this section to obtain an instance (G', k') of SPLIT CONTRACTION, where $k' = 2t + \lceil \delta t/k \rceil$, $k' \in O(k^2)$. Let $\mathsf{opt}(G')$ be the number of minimum edges that needs to be contracted in G' to convert it into a split graph.

By Lemma 5.2, there exists a set of edges F' in G' such that G'/F' is a split graph and the size of F' is at most k'. This implies that $\mathsf{opt}(G') \leq k'$. Let $\tilde{F}$ be the set of edges returned by algorithm $\mathcal{A}_\delta$ when the input is (G', k'). Since $\mathcal{A}_\delta$ returns an approximate solution within factor $(5/4 - \delta)$, we know that $|\tilde{F}| \leq (5/4 - \delta) \cdot \mathsf{opt}(G') \leq (5/4 - \delta) \cdot k'$. By Lemma 5.3, there exists a set of at most k/δ vertices in G that spans at least $(3\delta/2)t$ edges in G. Note that the proof of Lemma 5.3 can easily be converted into a polynomial time algorithm to obtain these sets of vertices and edges in G given F'. Since $\mathcal{A}_\delta$ is an FPT time approximation algorithm, it runs in time $f(k') \cdot |V(G')|^{O(1)} = f(k) \cdot |V(G)|^{O(1)}$ time.

We can conclude that there is an $f(k) \cdot |V(G)|^{O(1)}$-time algorithm such that, given an integer k and an edge colored graph G containing a colorful-k-clique, always outputs a vertex set of size at most k/δ vertices that span at least $2\gamma t$ colorful edges. Fix positive constants ϵ, α such that $0 < \epsilon < 1$; $1 < \alpha$ and $3 \leq 4\alpha\epsilon$. For $\delta = 1/\alpha$, the above conclusion contradicts Lemma 5.1. Hence, our assumption is wrong, which implies the correctness of the theorem. □

6 (NO) FPT APPROXIMATION ALGORITHM FOR CHORDAL CONTRACTION

In this section, we show that unlike CLIQUE CONTRACTION or SPLIT CONTRACTION, one can not obtain a lossy kernel of any size for CHORDAL CONTRACTION. It is known that for every $\alpha \geq 1$ and parameterized optimization problem Π, Π admits a fixed parameter tractable α-approximation algorithm if and only if Π has an α-approximate kernel [40, Proposition 2.11]. We prove that CHORDAL CONTRACTION parameterized by the solution size k, cannot be approximated within a factor of $F(k)$ in FPT time. Towards this we give a reduction from SET COVER and use a known result that no FPT time algorithm can approximate SET COVER within a factor of $F(k)$, where $F(k)$ is the function of k alone [35].

The parameterized optimization version of CHORDAL CONTRACTION is defined as follows.

$$\mathsf{CHC}(G, k, F) = \begin{cases} \min\{|F|, k+1\} & \text{if } G/F \text{ is a chordal graph} \\ \infty & \text{otherwise.} \end{cases}$$

Lokshtanov et al. [38] proved that CHORDAL CONTRACTION, parameterized by solution size k, does not admit a (classical) kernel of any size under widely believed assumption. They proved this by presenting a *parameter preserving* reduction from HITTING SET problem to CHORDAL CONTRACTION. In the following lemma, we argue that such *parameter preserving* reduction can also be obtained in case of the optimization version of problems. Our reduction is the same as the reduction given by Lokshtanov et al. [38]. We need some more arguments to show that there exists a 1-appt from SET COVER/k to CHORDAL CONTRACTION when parameterized by solution size. For the sake of completeness, we give the full reduction.
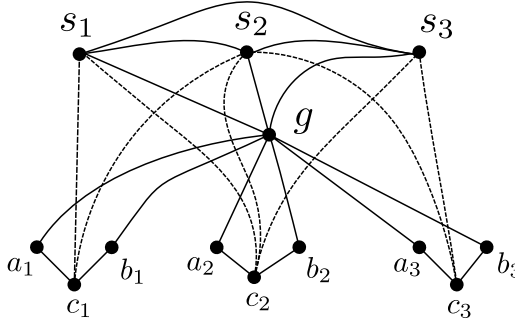
Fig. 6. Reduction from an instance $((U, S), k)$ of SET COVER to an instance (G, k) of CHORDAL CONTRACTION. Here $U = \{u_1, u_2, u_3\}$; $S = \{S_1, S_2, S_3\}$ where $S_1 = \{u_1, u_2\}$, $S_2 = \{u_1, u_2, u_3\}$, and $S_3 = \{u_2, u_3\}$. Straight lines (eg. a_1g) are part of gadget construction whereas dashed lines (eg. c_1s_1) are added because of containment relationship between elements and sets. Please refer to the Reduction Algorithm in the proof of Lemma 6.1.

LEMMA 6.1. *There exists a 1-approximate polynomial parameter transformation (1-appt) from* SET COVER$/k$ *to* CHORDAL CONTRACTION *parameterized by solution size.*

PROOF. To prove this lemma, we present a reduction algorithm, R_A, which given an instance $((U, S), k)$ of SET COVER$/k$ outputs an instance (G, k') of CHORDAL CONTRACTION. We also present a solution lifting algorithm that takes as input an instance $((U, S), k)$ of SET COVER$/k$; an output instance $(G, k') = R_A((U, S), k)$ of CHORDAL CONTRACTION; and a solution F to the instance (G, k'); and outputs a solution $\mathcal{F}$ to $((U, S), k)$ such that SC$/k((U, S), k, \mathcal{F})$ = CHC(G, k, F).

Without loss of generality, we assume that every element of the universe U is contained in some set S_i in S as otherwise $((U, S), k)$ is a trivial instance. We first present a reduction algorithm.

Reduction Algorithm : Given an instance $((U, S), k)$ of the SET COVER problem with $U = \{u_1, \cdots, u_n\}$ and $S = \{S_1, \cdots, S_m\}$, the algorithm constructs graph G as follows: It creates a vertex s_j for each set S_j in S and three vertices a_i, b_i and c_i for each element u_i in the universe U. It also adds a special vertex g to G. It adds following edges in G.

- an edge between any two different vertices corresponding to sets; (In other words, the algorithm converts set $\{s_1, s_2, \cdots, s_m\}$ into a clique by adding all edges $s_j s_{j'}$ for $1 \le j, j' \le m$ and $j \ne j'$.)
- edge gs_j for every j in $\{1, 2, \cdots, m\}$ and edges ga_i, gb_i for every i in $\{1, 2, \cdots, n\}$;
- edges $a_i c_i$ and $b_i c_i$ for every i in $\{1, 2, \cdots, n\}$;
- for an element x_i and a set S_j, if x_i is in S_j then it adds edge $c_i s_j$;

The algorithm returns (G, k) as an instance of CHORDAL CONTRACTION. See Figure 6 for an illustration.

It is easy to verify that graph G does not contain any induced cycle of length five or more. We have created cycles of length four for each element in the universe which intersects with each other only in g. Informally speaking, all these cycles can be killed by introducing the edge gc_j for every cycle. To introduce all these chords with at most k contractions, we need to carefully select at most k sets (and contract edges of the form gs_i) which *covers* all the elements. We argue that introducing chords of the form gs_j also kills other cycles of length four in the graph.

Solution Lifting Algorithm: Let F° be the given solution for (G, k). If $|F^\circ| \ge k + 1$ then the algorithm returns $\mathcal{F} = S$ as a solution. Otherwise, the algorithm first constructs another solution

for (G, k) with the following two operations. (a) If F° contains an edge of the form ga_i, gb_i, a_ic_i or b_ic_i, then replace it by gs_j, where S_j is any set containing u_i. (b) If F° contains an edge of the form s_jc_i, then replace it by gs_j. Let F be the solution obtained from F° by exhaustively applying these two operations. Consider a G/F-witness structure of G and let $W(g)$ be the witness set containing g. Define $\mathcal{F}$ as a collection of sets in $\mathcal{S}$ whose corresponding vertices are contained in $W(g)$. The algorithm returns $\mathcal{F}$ as a solution for $((U, \mathcal{S}), k)$.

We justify the two modification operations defined in solution lifting algorithm. It is easy to see that $|F| \leq |F^\circ|$. We slightly abuse notations and rename new vertex added while contracting edge gs_j as g. The only cycle affected by contracting edges of the form ga_i, gb_i, a_ic_i or b_ic_i is $\{g, a_i, c_i, b_i\}$. By contracting an edge of the form gs_j where S_j is any set containing u_i, we introduce another chord gc_i which *destroys* the cycle. Similarly, if F° contains an edge of the form s_jc_i, then the only four-cycles of G that gets a chord in $G/\{s_jc_i\}$ are: $\{g, a_i, c_i, b_i\}$, $\{s_j, g, a_i, c_i\}$, and $\{s_j, g, b_i, c_i\}$. All of these cycles get a chord when the edge gs_j is contracted instead. This implies that if G/F° is a chordal graph then so is G/F.

We now prove that the cycles present in the graph G are of a very specific type. This claim is similar to Proposition 1 in [38]. Note that because of Claim 1, to convert G into a chordal graph it is sufficient to introduce chords gc_i for every i in $\{1, 2, \cdots, n\}$.

Claim 1. Graph G does not contain any induced cycle of length five or more. The only induced cycles of length four in the graph G are of one of the three forms: (i) $\{g, a_i, c_i, s_j\}$ for some element u_i and set S_j containing it; (ii) $\{g, b_i, c_i, s_j\}$ for some element u_i and set S_j containing it; (iii) $\{g, a_i, c_i, b_i\}$ for some element u_i.

Proof: We define subsets T, A, B, C of $V(G)$ as collections of vertices s_j's, a_i's, b_i's and c_i's respectively. Formally, $T = \{s_1, s_2, \cdots, s_m\}$; $A = \{a_1, a_2, \cdots, a_n\}$; $B = \{b_1, b_2, \cdots, b_n\}$ and $C = \{c_1, c_2, \cdots, c_n\}$. Note that $G[T]$ is a clique whereas A, B, C are independent sets in G.

Since $G[T \cup \{g\}]$ is a clique, any induced cycle of length at least four contains at most two vertices of $T \cup \{g\}$. As $G \setminus (T \cup \{g\})$ is a collection of induced paths on three vertices, and hence acyclic, the largest induced cycle possible in G is of length five. We note that every induced paths is of the form $\{a_i, c_i, b_i\}$ and only other vertex adjacent to a_i, b_i is g. Hence, such path can not be part of an induced C_5. This implies that G does not contain an induced cycle of length five or more.

Assume that there exists an induced C_4 with two vertices, say $s_j, s_{j'}$, in T. By construction, the only vertices which are adjacent with $s_j, s_{j'}$ are in the set $C \cup \{g\}$. Since g is adjacent with both these vertices, it can not be part of induced C_4 that contains $s_j, s_{j'}$. This implies that the remaining two vertices in C_4 are from C. As $s_j, s_{j'}$ are adjacent to each other, the remaining two vertices in C_4 must be adjacent to each other. This contradicts the fact that C is an independent set in G. Hence our assumption is wrong and no such induced C_4 exists.

Consider an induced C_4 which contains exactly one vertex, say s_j, in T. Assume that this induced C_4 does not contain g. By construction, only neighbors of s_j outside T are in C. Let c_i, c_j are two vertices contained in this induced C_4. Since the only common neighbor of c_i, c_j outside T is g, our assumption that this C_4 does not contain g is wrong. This implies every such induced C_4 contains g. Since the only other neighbor of s_j is in C, one of the remaining vertex in this cycle is from set C. Let that vertex be c_i. Since a_i or b_i are the only common neighbors of g and c_i, the only possible cycles are of the form (i) or (ii) mentioned in the claim. As there is no edge between following pairs of vertices- (g, c_i), (s_j, a_i) and (s_j, b_i), this cycle is indeed an induced C_4.

Consider an induced C_4 which does not intersect with T. Since $G \setminus (T \cup \{g\})$ is acyclic, every cycle of this type contains g. In this cycle, neighbors of g are from sets A and B. By construction, vertices a_i and $b_{i'}$ have a common neighbor only if $i = i'$. This implies the only possible induced C_4 which does not intersect with T is of the form (iii).

As we have considered all cases exhaustively, this proves the claim.   ◇

We now prove that any solution of SET COVER/k naturally leads to a solution for CHORDAL CONTRACTION. For any subset $\mathcal{F}$ of $\mathcal{S}$, let $F_{\mathcal{F}}$ be the set of edges in G which are incident on g and s_i for some s_i in $\mathcal{F}$.

Claim 2: If $\mathcal{F}$ is a set cover of instance the $((U, \mathcal{S}), k)$, then $G/F_{\mathcal{F}}$ is a chordal graph.
Proof: Let H be the graph obtained from G by contracting all edges in $F_{\mathcal{F}}$. Since $\mathcal{F}$ covers all the elements of U, contracting all the edges in $F_{\mathcal{F}}$ introduces edge gc_i for every i in $\{1, 2, \cdots, n\}$ in graph H. By Claim 1, all the induced cycles in G are of the form $\{g, a_i, c_i, s_j\}$ or $\{g, b_i, c_i, s_j\}$ or $\{g, a_i, c_i, b_i\}$ for some element u_i. Hence, there is no induced cycle of length four or more in H; thus it is a chordal graph.   ◇

Let F be the given solution for (G, k) such that G/F is a chordal graph and $|F|$ is at most k. Let $\mathcal{F}$ be the solution for SET COVER/k instance returned by the solution lifting algorithm. Recall that the modification operation (a) or (b) mentioned in the solution lifting algorithm does not change the size of F.

Claim 3: $\mathcal{F}$ is a set cover of size at most $|F|$ for $(U, \mathcal{S})$.
Proof: Consider the four-cycle given by $\{g, a_i, c_i, b_i\}$ in graph G. Since G/F is a chordal graph, there exists an edge gc_i in $E(G/F)$. This implies there is an edge gs_j in F for some set S_j which contains u_i. Hence there is a set S_j in $\mathcal{F}$ which contains element u_i. Since this is true for any element in U, $\mathcal{F}$ is a set cover for $(U, \mathcal{S})$. As $|W(g)|$ is at most $|F| + 1$ and it contains vertex g, upper bound on $\mathcal{F}$ follows.   ◇

We are now in the position to conclude the proof. Claim 2 implies that $\text{OPT}_{\text{CHC}(G,k)} \le \text{OPT}_{\text{SC}/k}((U, \mathcal{S}), k)$. Moreover, if $|F| \ge k + 1$, then solution lifting algorithm returns $\mathcal{F} = \mathcal{S}$. In this case, $\text{CHC}(G, k, F) = k + 1 = \text{SC}/k((U, \mathcal{S}), k, \mathcal{S})$. If $|F| \le k$ and G/F is a chordal graph then by Claim 3, $\text{SC}/k((U, \mathcal{S}), k, \mathcal{F}) \le \text{CHC}(G, k, F)$. This implies that there exists a 1-appt from SET COVER/k to CHORDAL CONTRACTION when parameterized by solution size.   □

Karthik et al. [35, see conclusion] showed that assuming FPT$\ne$ W[1], no FPT time algorithm can approximate SET COVER within a factor of $F(k)$. Pipelining this result with Lemma 6.1, we get the following result.

THEOREM 1.4. *Assuming FPT$\ne$ W[1], no FPT time algorithm can approximate* CHORDAL CONTRACTION *within a factor of $F(k)$. Here, $F(k)$ is a function depending on k alone.*

7 CONCLUSION

In this paper, we studied the $\mathcal{F}$-CONTRACTION problem, where $\mathcal{F}$ is a subfamily of chordal graphs, in the realm of parameterized approximation. We showed that CLIQUE CONTRACTION admits a PSAKS. On the other hand, SPLIT CONTRACTION admits a factor $(2 + \epsilon)$-FPT-approximation algorithm, for any $\epsilon > 0$. In fact, we showed that for any $\epsilon > 0$, SPLIT CONTRACTION admits an $(2 + \epsilon)$-approximate kernel with $O(k^{f(\epsilon)})$ vertices. We complemented this result by showing that, assuming Gap-ETH, no FPT time algorithm can approximate SPLIT CONTRACTION within a factor of $\left(\frac{5}{4} - \delta\right)$, for any fixed constant $\delta > 0$. Finally, we showed that, assuming FPT$\ne$ W[1], CHORDAL CONTRACTION does not admit any $F(k)$-FPT-approximation algorithm. Our results add to this growing list of collection of FPT-approximable and FPT-in-approximable problems.

Our paper also shows that further classification of problems using lossy kernels are of interest and could shed new light on even well-studied problems. The paper naturally leads to the following

question: can the gap between upper and lower bounds for SPLIT CONTRACTION brought closer? We conjecture that SPLIT CONTRACTION does admit $\frac{5}{4}$-FPT-approximation algorithm.

ACKNOWLEDGMENTS

REFERENCES

[1] Akanksha Agarwal, Saket Saurabh, and Prafullkumar Tale. 2017. On the Parameterized Complexity of Contraction to Generalization of Trees. *Theory of Computing Systems* (2017), 1–28.

[2] Akanksha Agrawal, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. 2017. Split Contraction: The Untold Story. In *STACS*. 5:1–5:14.

[3] Takao Asano and Tomio Hirata. 1983. Edge-contraction problems. *J. Comput. System Sci.* 26, 2 (1983), 197–208.

[4] Reuven Bar-Yehuda and Shimon Even. 1981. A linear-time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms* 2, 2 (1981), 198–203.

[5] Ivan Bliznets, Fedor V. Fomin, Marcin Pilipczuk, and Michal Pilipczuk. 2014. A subexponential parameterized algorithm for proper interval completion. In *ESA*. 173–184.

[6] Ivan Bliznets, Fedor V. Fomin, Marcin Pilipczuk, and Michal Pilipczuk. 2016. A subexponential parameterized algorithm for interval completion. In *SODA*. 1116–1131.

[7] Leizhen Cai. 1996. Fixed-parameter tractability of graph modification problems for hereditary properties. *Inform. Process. Lett.* 58, 4 (1996), 171–176.

[8] Leizhen Cai and Chengwei Guo. 2013. Contracting Few Edges to Remove Forbidden Induced Subgraphs. In *IPEC*. 97–109.

[9] Yixin Cao. 2015. Unit interval editing is Fixed-parameter tractable. In *ICALP*. 306–317.

[10] Yixin Cao. 2016. Linear recognition of almost interval graphs. In *SODA*. 1096–1115.

[11] Yixin Cao and Dániel Marx. 2014. Chordal Editing is Fixed-Parameter Tractable. In *STACS*. 214–225.

[12] Yixin Cao and Dániel Marx. 2015. Interval deletion is fixed-parameter tractable. *ACM Transactions on Algorithms (TALG)* 11, 3 (2015), 21.

[13] Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. 2017. From Gap-ETH to FPT-Inapproximability: Clique, Dominating Set, and More. In *FOCS*. 743–754.

[14] Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. 2017. From Gap-ETH to FPT-Inapproximability: Clique, Dominating Set, and More. *CoRR* abs/1708.04218 (2017).

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

[16] Reinhard Diestel. 2012. *Graph Theory, 4th Edition*. Graduate texts in mathematics, Vol. 173. Springer.

[17] Rod G. Downey and Michael R. Fellows. 2013. *Fundamentals of Parameterized complexity*. Springer-Verlag.

[18] Pål Grønås Drange, Markus Sortland Dregi, Daniel Lokshtanov, and Blair D Sullivan. 2015. On the threshold of intractability. In *Algorithms-ESA 2015*. Springer, 411–423.

[19] Pål Grønås Drange, Fedor V Fomin, Michał Pilipczuk, and Yngve Villanger. 2014. Exploring Subexponential Parameterized Complexity of Completion Problems. In *STACS*. 288–299.

[20] Pål Grønås Drange and Michal Pilipczuk. 2018. A Polynomial Kernel for Trivially Perfect Editing. *Algorithmica* 80 (2018), 3481–3524.

[21] Pavel Dvorák, Andreas Emil Feldmann, Dusan Knop, Tomás Masarík, Tomas Toufar, and Pavel Veselý. 2018. Parameterized Approximation Schemes for Steiner Trees with Small Number of Steiner Vertices. In *STACS*. 26:1–26:15.

[22] Eduard Eiben, Danny Hermelin, and M. S. Ramanujan. 2017. Lossy Kernels for Hitting Subgraphs. In *MFCS*. 67:1–67:14.

[23] Eduard Eiben, Mithilesh Kumar, Amer E Mouawad, Fahad Panolan, and Sebastian Siebertz. 2019. Lossy kernels for connected dominating set on sparse graphs. *SIAM Journal on Discrete Mathematics* 33, 3 (2019), 1743–1771.

[24] Jörg Flum and Martin Grohe. 2006. *Parameterized Complexity Theory*. Springer.

[25] Fedor V. Fomin, Stefan Kratsch, Marcin Pilipczuk, Michał Pilipczuk, and Yngve Villanger. 2014. Tight bounds for parameterized complexity of cluster editing with a small number of clusters. *J. Comput. System Sci.* 80, 7 (2014), 1430–1447.

[26] Fedor V Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. 2019. *Kernelization: theory of parameterized preprocessing.* Cambridge University Press.

[27] Fedor V. Fomin and Yngve Villanger. 2013. Subexponential parameterized algorithm for minimum fill-in. *SIAM J. Comput.* 42, 6 (2013), 2197–2216.

[28] Esha Ghosh, Sudeshna Kolay, Mrinal Kumar, Pranabendu Misra, Fahad Panolan, Ashutosh Rai, and M. S. Ramanujan. 2015. Faster parameterized algorithms for deletion to split graphs. *Algorithmica* 71, 4 (2015), 989–1006.

[29] Petr A Golovach, Pim van 't Hof, and Daniël Paulusma. 2013. Obtaining planarity by contracting few edges. *Theoretical Computer Science* 476 (2013), 38–46.

[30] Martin Charles Golumbic. 2004. *Algorithmic graph theory and perfect graphs.* Vol. 57. Elsevier.

[31] Sylvain Guillemot and Dániel Marx. 2013. A faster FPT algorithm for Bipartite Contraction. *Inform. Process. Lett.* 113, 22–24 (2013), 906–912.

[32] Chengwei Guo and Leizhen Cai. 2015. Obtaining split graphs by edge contraction. *Theoretical Computer Science* 607 (2015), 60–67.

[33] Pinar Heggernes, Pim van 't Hof, Benjamin Lévêque, Daniel Lokshtanov, and Christophe Paul. 2014. Contracting Graphs to Paths and Trees. *Algorithmica* 68, 1 (2014), 109–132.

[34] Pinar Heggernes, Pim van't Hof, Daniel Lokshtanov, and Christophe Paul. 2013. Obtaining a Bipartite Graph by Contracting Few Edges. *SIAM Journal on Discrete Mathematics* 27, 4 (2013), 2143–2156.

[35] Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. 2018. On the parameterized complexity of approximating dominating set. In *STOC*. ACM, 1283–1296.

[36] R Krithika, Diptapriyo Majumdar, and Venkatesh Raman. 2018. Revisiting connected vertex cover: FPT algorithms and lossy kernels. *Theory of Computing Systems* 62, 8 (2018), 1690–1714.

[37] R Krithika, Pranabendu Misra, Ashutosh Rai, and Prafullkumar Tale. 2016. Lossy Kernels for Graph Contraction Problems. In *FSTTCS*. 23:1–23:14.

[38] Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. 2013. On the hardness of eliminating small induced subgraphs by contracting edges. In *IPEC*. 243–254.

[39] Daniel Lokshtanov, Pranabendu Misra, Fahad Panolan, Geevarghese Philip, and Saket Saurabh. 2020. A $(2 + \epsilon)$-factor Approximation Algorithm for Split Vertex Deletion. In *47th International Colloquium on Automata, Languages and Programming, ICALP 2020 (LIPIcs, Vol. 168).* to appear.

[40] Daniel Lokshtanov, Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. 2017. Lossy kernelization. In *STOC*. 224–237.

[41] Pasin Manurangsi. 2019. A Note on Max k-Vertex Cover: Faster FPT-AS, Smaller Approximate Kernel and Improved Approximation. In *SOSA*.

[42] Moni Naor, Leonard J Schulman, and Aravind Srinivasan. 1995. Splitters and near-optimal derandomization. In *FOCS*. 182–191.

[43] Rolf Niedermeier. 2006. *Invitation to fixed-parameter algorithms.* Oxford University Press.

[44] M. S. Ramanujan. 2019. An Approximate Kernel for Connected Feedback Vertex Set. In *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany (LIPIcs, Vol. 144).* 77:1–77:14.

[45] Sebastian Siebertz. 2017. Lossy kernels for connected distance-r domination on nowhere dense graph classes. *arXiv preprint arXiv:1707.09819* (2017).

[46] René van Bevern, Till Fluschnik, and Oxana Yu Tsidulko. 2018. On $(1 + \epsilon)$-approximate problem kernels for the Rural Postman Problem. *arXiv preprint arXiv:1812.10131* (2018).

[47] Toshimasa Watanabe, Tadashi Ae, and Akira Nakamura. 1981. On the removal of forbidden graphs by edge-deletion or by edge-contraction. *Discrete Applied Mathematics* 3, 2 (1981), 151–153.

[48] Toshimasa Watanabe, Tadashi Ae, and Akira Nakamura. 1983. On the NP-hardness of edge-deletion and-contraction problems. *Discrete Applied Mathematics* 6, 1 (1983), 63–78.