Polynomial Kernels for Vertex Cover Parameterized by Small Degree Modulators*

Diptapriyo Majumdar, Venkatesh Raman, and Saket Saurabh

The Institute of Mathematical Sciences, HBNI, Chennai, India.
{diptapriyom|vraman|saket}@imsc.res.in

Abstract

Vertex Cover is one of the most well studied problems in the realm of parameterized algorithms. It admits a kernel with $\mathcal{O}(\ell^2)$ edges and 2ℓ vertices where ℓ denotes the size of the vertex cover we are seeking for. A natural question is whether Vertex Cover is fixed-parameter tractable or admits a polynomial kernel with respect to a parameter k, that is, provably smaller than the size of the vertex cover. Jansen and Bodlaender [STACS 2011, TOCS 2013] raised this question and gave a kernel for Vertex Cover of size $\mathcal{O}(f^3)$, where f is the size of a feedback vertex set of the input graph. We continue this line of work and study Vertex Cover with respect to a parameter that is always smaller than the solution size and incomparable to the size of the feedback vertex set of the input graph. Our parameter is the number of vertices whose removal results in a graph of maximum degree two. While vertex cover with this parameterization can easily be shown to be fixed-parameter tractable (FPT), we show that it has a polynomial kernel.

The input to our problem consists of an undirected graph G, $S \subseteq V(G)$ such that $|S| = k$ and $G[V(G) \setminus S]$ has maximum degree at most two and a positive integer ℓ. Given (G, S, ℓ), in polynomial time we output an instance (G', S', ℓ') such that $|V(G')|$ is $\mathcal{O}(k^5)$, $|E(G')|$ is $\mathcal{O}(k^6)$ and G has a vertex cover of size at most ℓ if and only if G' has a vertex cover of size at most ℓ'.

When $G[V(G) \setminus S]$ has maximum degree at most one, we improve the known kernel bound from $\mathcal{O}(k^3)$ vertices to $\mathcal{O}(k^2)$ vertices (and $\mathcal{O}(k^3)$ edges). More generally, given (G, S, ℓ) where every connected component of $G \setminus S$ is a clique of at most d vertices (for constant d), in polynomial time, we output an equivalent instance (G', S', ℓ') for the same problem where $|V(G')|$ is $\mathcal{O}(k^d)$. We also show that for this problem, when $d \geq 3$, a kernel with $\mathcal{O}(k^{d-\varepsilon})$ bits cannot exist for any $\varepsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

1 Introduction and Motivation

In the early years of parameterized complexity and algorithms, problems were almost always parameterized by the solution size. Recent research has focussed on other parameterizations based on structural parameters in the input [13], or above or below some tight guaranteed values [19, 20, 28, 6, 25, 12]. The reasons are many. Such 'non-standard' parameters are more likely to be small in practice. Also, once a problem is shown to be fixed-parameter tractable (FPT) (see Section 2 for definitions) or to have a polynomial kernel by a parameterization, it is natural to ask whether the problem is FPT (and admits a polynomial kernel) when parameterized by a parameter that is provably smaller. In the same vein, if we show that a problem is W-hard under a parameterization, it is natural to ask whether it is FPT when parameterized by a parameter that is potentially larger.

One of the earliest papers in the realm of such alternate parameterization dates back to 1981. Let $\mathfrak{O}_k$ denote the set of all graphs G such that the length of the longest odd cycle is upper bounded by k.

*A preliminary version of this paper [29] appeared in the proceedings of the 10th International Symposium on Parameterized and Exact Computation (IPEC 2015).

Hsu et al. [22] initiated a study of NP-hard optimization problems on $\mathfrak{O}_k$. In particular, they showed that a maximum sized independent set on a graph in $\mathfrak{O}_k$ on n vertices can be found in time $n^{\mathcal{O}(k)}$. Later, Grötschel and Nemhauser [18] did a similar study for Max-Cut and obtained an algorithm with running time $n^{O(k)}$ on a graph in $\mathfrak{O}_k$ on n vertices. These algorithms, using modern techniques, can be made FPT and also shown to not admit polynomial kernels unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [31]. Later Cai [4] did a similar study for Coloring problems. Fellows et al. [14] studied alternate parameterizatons for problems that were proven to be intractable with respect to standard parameterizations. This led to the whole new ecology program and opened up a floodgate of new and exciting research. We refer to Fellows et al. [13] for a detailed introduction to the whole program. The kernelization results tend to be harder in this framework, as the rules need to capture the interaction of the structural parameter with the rest of the input, against simply using the property of feasible solutions when 'solution size' is used as the parameter.

A *vertex cover* in a graph is a set of vertices that contains at least one endpoint from every edge of the graph. In the Vertex Cover problem, given an undirected graph G and an integer ℓ, the task is to determine whether G has a vertex cover of size at most ℓ. Vertex Cover is one of the most well studied problems in the realm of parameterized algorithms and admits an algorithm with running time $1.2738^{\ell} n^{\mathcal{O}(1)}$ and a kernel with $\mathcal{O}(\ell^2)$ edges and 2ℓ vertices [5, 30]. The set S is also called *vertex cover* of the graph. A natural question is whether Vertex Cover admits a polynomial kernel (or a parameterized algorithm) with respect to a parameter k, that is, provably smaller than the size of the vertex cover. Jansen and Bodlaender [23] first raised this question and since then the following are some of the results known for Vertex Cover under different parameterizations.

- Vertex Cover admits a kernel of size $\mathcal{O}(f^3)$, where f is the size of a feedback vertex set of the input graph [23].

- For Vertex Cover parameterized by the size of an odd cycle transversal and König vertex deletion set, there is a randomized polynomial kernel [27]. Here an *odd cycle transversal* is a set of vertices whose deletion makes the resulting graph bipartite. A *König vertex deletion set* is a set of vertices whose deletion makes the resulting graph into a *König graph.* A graph is said to be a *König graph* if the size of its minimum vertex cover exactly equals the size of its maximum matching. This result has recently been strengthened by Kratsch [26] who parameterized above even a stronger lower bound on the solution, $2\mathsf{LP} - \mu(\mathsf{G})$, where LP denotes the optimum value of the standard LP relaxation and $\mu(\mathsf{G})$ denotes the size of a maximum matching of the graph G.

- Vertex Cover parameterized by the deletion set to chordal graphs or perfect graphs has no polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [2, 13]. A graph is said to be a chordal graph if it has no induced cycle of length at least four. A perfect graph is a graph in which the chromatic number of every induced subgraph equals the size of a largest clique of that subgraph.

- Vertex Cover parameterized by treewidth-2-modulator (i.e. when $G[V(G) \setminus S]$ is a graph of treewidth at most 2) has no polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [8]. In fact Vertex Cover does not have polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ when it is parameterized by the size of a modulator to an *outer-planar graph* or even to a *mock-forest* [16]. A mock-forest is a graph every vertex of which is contained in at most one cycle. It is easy to see that the size of a degree-2-modulator is larger than all these parameters (discussed in this item) for which a polynomial kernel is unlikely to exist. Also, the size of a degree-2-modulator is incomparable with the size of a feedback vertex set for which there is a cubic kernel for Vertex Cover (see Figure 1 for a hierarchy of parameters considered in this paper; some of the parameters are explained in the description of our results and related work later in this section).

In this paper we continue this line of work on Vertex Cover and study it with respect to a parameter that is always smaller than the solution size and incomparable to the feedback vertex set of the input graph. In particular, we consider the Vertex Cover problem parameterized by the number of vertices whose removal results in a graph of maximum degree at most d, where $d \geq 1$.

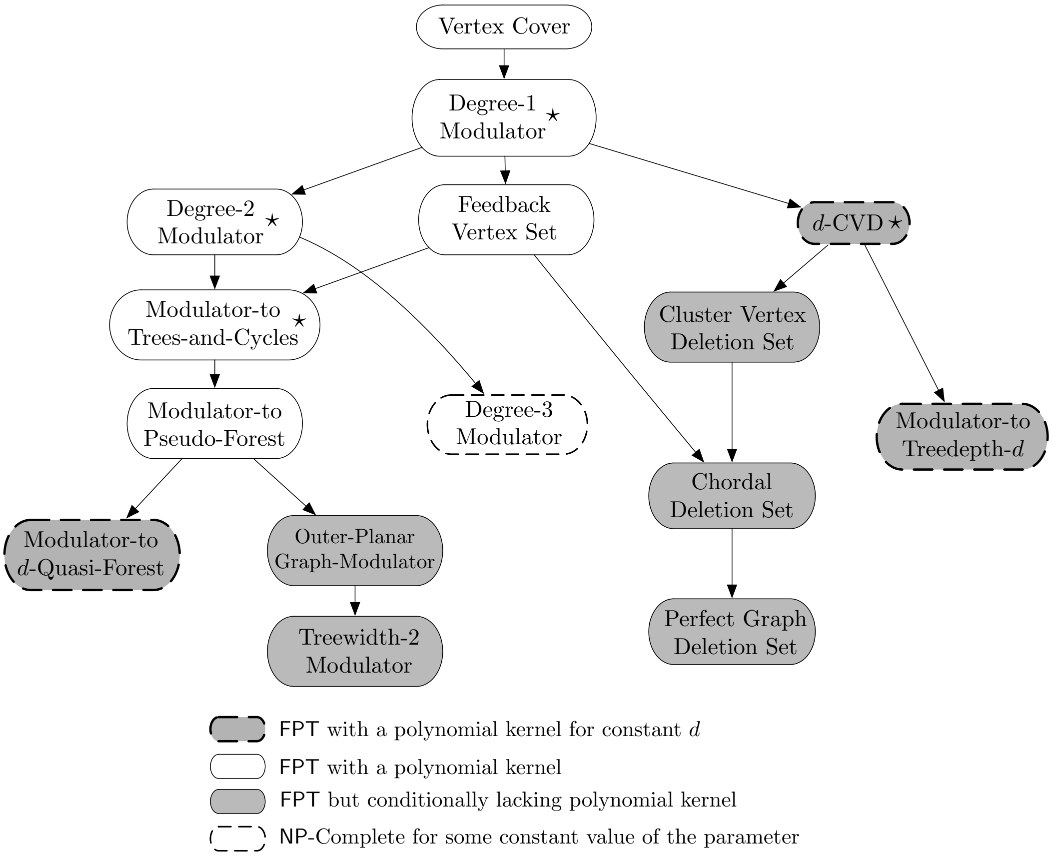

Figure 1: Hierarchy of parameters. An arrow from parameter a to parameter b indicates that the minimum value of parameter b is at most the minimum value of parameter a. Parameters marked $\star$ indicate the ones considered in this paper.

Vertex Cover parameterized by degree-d-Modulator (VC-d-Mod)
Input: An undirected graph G, $S \subseteq V(G)$ of size at most k such that $G[V(G) \setminus S]$ is a graph of degree at most d and an integer ℓ.
Parameter: k
Question: Does G have a vertex cover of size at most ℓ?

Vertex Cover is known to be NP-Complete even on graphs of maximum degree 3 and thus the d in VC-d-Mod must be upper bounded by 2, else we cannot even hope to have an algorithm of the form $n^{f(k)}$ for any function f. On the other hand Vertex Cover is polynomial time solvable when the maximum degree is at most 2 (or in other words, when every component of the graph is either an isolated vertex or a path or a cycle).

Let G be the input graph along with a vertex subset S such that $|S| \leq k$ and $G[V(G) \setminus S]$ has maximum degree at most 2. We call S a *degree-2-modulator of the graph*. By 'guessing' (i.e. trying all possible choices for) the intersection of S with the optimal vertex cover, and solving the remaining problem in polynomial time, we can find a minimum vertex cover of G in $\mathcal{O}(2^k n^{\mathcal{O}(1)})$ time. This shows that VC-2-Mod is FPT. One of our main results is a polynomial kernel for VC-2-Mod.

Our Results.

- We obtain a kernel for VC-2-Mod with $\mathcal{O}(k^5)$ vertices, and $\mathcal{O}(k^6)$ edges. Strømme [34] independently obtained a kernel with $\mathcal{O}(k^7)$ vertices for this problem. Note that a graph with degree at most 2 has treewidth at most 2. Hence, our result is in contrast to the fact that

Vertex Cover parameterized by treewidth-2-modulator (i.e. when $G[V(G) \setminus S]$ is a graph of treewidth at most 2) has no polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [8].

- We extend our idea for VC-2-Mod to provide a kernel with $\mathcal{O}(k^9)$ vertices when k is the number of vertices whose deletion results in a graph that is a disjoint union of trees and cycles.
- We also address the kernelization question for VC-1-Mod. Here, a kernel with $\mathcal{O}(k^3)$ vertices was already known from the result by Jansen and Bodlaender [23] for Vertex Cover parameterized by the feedback vertex set size. This follows as the size of the feedback vertex set is at most the size of a degree-1-modulator. We improve the kernel size to $\mathcal{O}(k^2)$ vertices.
- More generally, we consider the Vertex Cover problem when parameterized by the size of a subset of vertices whose removal results in a graph with all components being cliques of size at most a constant d. We call a graph G *d-cluster graph* if every connected component of G is a clique and has size at most d. In particular we study the following problem

  Vertex Cover parameterized by d-CVD (VC-param-d-CVD)
  Input: An undirected graph G, $S \subseteq V(G)$ of size at most k such that every connected component of $G[V(G) \setminus S]$ is a clique with at most d vertices and an integer ℓ.
  Parameter: k
  Question: Does G have a vertex cover of size at most ℓ?

  Observe that VC-1-Mod and VC-param-2-CVD are the same problems. It is known that if the resulting graph is simply a clique (with no bound on the size), then a polynomial kernel is unlikely [2].

  For VC-param-d-CVD, we provide a kernel with $\mathcal{O}(k^d)$ vertices. At an intermediate step of the kernel, we reduce the problem to an equivalent hypergraph problem. We believe that this idea of using hyperedges to capture certain constraints could find applications while developing a compression for a parameterized problem. We also show that for any $d \geq 3, \varepsilon > 0$, a kernel of $\mathcal{O}(k^{d-\varepsilon})$ size is unlikely unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

Observe that we have always assumed that the *modulator* is given as a part of the input. However, this constraint can be relaxed as both VC-2-MOD and VC-param-d-CVD admit constant factor approximation algorithms. Constant factor approximation algorithms can be obtained by greedily finding an obstruction (like a vertex v with degree at least three and any of its three neighbors in the case of VC-2-MOD) and selecting all the vertices in this obstruction to the approximate solution we are constructing. For example, for VC-2-MOD, this results in a factor 4-approximation algorithm and for VC-param-d-CVD, an approximation algorithm with factor $(d+1)$. Thus, if the modulators are not available as a part of the input, we can first compute them using the polynomial time constant factor approximation algorithms and then run our kernelization algorithms using these modulators. These will result in kernels with the same asymptotic upper bounds as mentioned above.

Related Work: Since the first appearance of our result on VC-2-Mod [29], Fomin and Strømme [17] considered Vertex Cover parameterized by two related parameters: distance to a pseudo-forest (a graph in which each component has at most one cycle), VC-Psuedo-Forest and to a mock-forest, VC-Mock-Forest. They gave a $O(k^{12})$ kernel for VC-Psuedo-Forest and proved that no polynomial kernel is possible unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ for VC-Mock-Forest. Hols and Kratsch [21] considered a parameter that generalizes our parameters and those of Fomin and Strømme [17] and showed a kernel for Vertex Cover with $\mathcal{O}(k^{3d+9})$ vertices where k is the number of vertices whose deletion results in a *d-quasi-forest*. A graph is said to be a d-quasi-forest when each of its components has a feedback vertex set with at most d vertices. Note that a graph with maximum degree 2 is a 1-quasi-forest, and hence a degree-2-modulator is a modulator to a 1-quasi-forest. So the result due to Hols and Kratsch [21] provides a kernel for VC-2-Mod with $\mathcal{O}(k^{12})$ vertices, while we give a kernel with $O(k^5)$ vertices. Hols and Kratsch also observe that a modulator to a $(d+2)$-cluster graph is a modulator to a d-quasi-forest and hence our lower bound result for VC-param-$(d+2)$-CVD provides a lower bound result for Vertex Cover parameterized by a modulator to d-quasi-forest.

Parameterization Considered	Polynomial Kernel	Lower Bound
Solution Size	$2k - c \cdot \log k$ vertices [28] k^2 edges [7]	$\Omega(k^2)$ size [9]
VC-1-Mod	$\mathcal{O}(k^3)$ vertices [23]	$\Omega(k^2)$ size [9]
Feedback Vertex Set	$\mathcal{O}(k^3)$ vertices [23]	$\Omega(k^2)$ size [9]
VC-1-Mod	$\mathcal{O}(k^2)$ vertices $\star$	$\Omega(k^2)$ size [9]
VC-2-Mod	$\mathcal{O}(k^7)$ vertices [34]	$\Omega(k^3)$ size $\star$
VC-2-Mod	$\mathcal{O}(k^5)$ vertices $\star$	$\Omega(k^3)$ size $\star$
Cluster Vertex Deletion	No [2]	–
VC-param-d-CVD	$\mathcal{O}(k^d)$ vertices $\star$	$\Omega(k^d)$ size $\star$
Treewidth-2-Modulator	No [8]	–
VC-Pseudo-Forest	$\mathcal{O}(k^{12})$ vertices [16]	$\Omega(k^3)$ size $\star$
VC-Trees-Cycles	$\mathcal{O}(k^9)$ vertices $\star$	$\Omega(k^3)$ size $\star$
VC-d-Quasi-Forest	$\mathcal{O}(k^{3d+9})$ [21]	$\Omega(k^{d+2})$ size $\star$
VC-Mock-Forest	No [16]	–
Treedepth-d-Modulator	$\mathcal{O}(k^{2^{\mathcal{O}(d^2)}})$ size [3]	$\Omega(k^d)$ size $\star$

Table 1: Summary of Results: Results marked $\star$ indicate our results

Bougeret and Sau [3] considered a related parameter which is the distance to a graph with treedepth at most d, and showed a kernel of size $\mathcal{O}(k^{2^{\mathcal{O}(d^2)}})$ for Vertex Cover; here k is the size of a set whose deletion results in a graph with treedepth at most d (see [3] for the definition of treedepth). It turns out that a clique with d vertices has treedepth d. So the parameter which is the distance to a graph with treedepth at most d is a generalization of our parameterization in VC-param-d-CVD. Hence, the result due to Bougeret and Sau [3] provides a kernel for VC-param-d-CVD of size $\mathcal{O}(k^{2^{\mathcal{O}(d^2)}})$ while our bound is $O(k^d)$. Similarly for the same reason, our lower bound result for VC-param-d-CVD provides a lower bound result to Vertex Cover parameterized by treedepth-d-modulator. The focus in Bougeret and Sau's [3] and Hols and Kratsch's [21] results are to show the existence of a polynomial kernel for a more general parameter (rather than obtaining the smallest possible size) while our focus is on giving as small a kernel as possible for the special parameters. See Table 1 for a summary of these results.

Organization of the rest of the paper. In the next section, we give definitions and notations in parameterized complexity including kernelization. We also state the expansion lemma used in several of our kernel results. In Section 3, we develop our polynomial kernel for VC-2-Mod . In Section 4, we give a polynomial kernel for vertex cover parameterized by a modulator whose removal results in a graph whose each component is a tree or a cycle. In Section 5, we consider VC-param-d-CVD and give polynomial kernels for each fixed d. In Section 6, we provide conditional lower bounds for the size of the kernel for the problems considered in the paper.

2 Preliminaries and Definitions

We use $\mathbb{N}$ to denote the set of all natural numbers. For $r \in \mathbb{N}$, by $[r]$ we denote the set $\{1, 2, \ldots, r\}$. Given a set S, we use $\binom{S}{r}$ to denote the collection of all subsets of S with exactly r elements. We use $\binom{S}{\leq r}$ to denote the collection of all subsets of S with at most r elements. We use $\binom{S}{\geq r}$ to denote the collection of all subsets of S with at least r elements. We use standard graph theoretic terminology from the book of Diestel [10] for the graph-related terms that are not explicitly defined here. We consider only finite undirected graphs. We use n to denote the number of vertices and m to denote the number of edges of the graph. For a graph $G = (V, E)$, we use $V(G)$ to denote the set of vertices and $E(G)$ to denote the set of edges. For a set $X \subseteq V(G)$, *the subgraph of G induced by X* is denoted by $G[X]$ and it is defined as the subgraph of G with vertex set X and edge set $\{(u, v) \in E(G)|u, v \in X\}$. Given a graph $G = (V, E)$ and a subset $X \subseteq V(G)$, we use $G \setminus X$ to denote the graph induced by $V(G) \setminus X$. For a vertex $v \in V(G)$, we use $deg_G(v)$ to denote the degree of the vertex v in G. All vertices adjacent to a vertex v are called neighbors of v and the set of all such vertices is called the neighborhood of v. The neighborhood of v in G is denoted by $N_G(v)$. When the graph is clear from the context, we omit the subscript. Throughout the paper we denote the *vertex cover number* (the size of a minimum vertex cover) of G by $\mathsf{vc}(G)$.

Definition 2.1 (Fixed-Parameter Tractability)**.** *A parameterized problem Π is a subset of $\Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet. We say that a parameterized problem Π is* fixed-parameter tractable (or FPT)*, if there is an algorithm solving the problem Π, that on input (x, k) runs in time $f(k)|x|^{\mathcal{O}(1)}$, where $f : \mathbb{N} \mapsto \mathbb{N}$ is an arbitrary computable function and $|x|$ is the length of the input x.*

Definition 2.2 (Kernelization)**.** *Let $\Pi \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem. Kernelization is a polynomial time procedure that replaces the input instance (I, k) by a reduced instance (I', k') such that*

- *$|I'| + k' \leq g(k)$ for some function g depending only on k, and*
- *$(I, k) \in \Pi$ if and only if $(I', k') \in \Pi$.*

If $g(k) \in k^{\mathcal{O}(1)}$, then we say that Π has a polynomial kernel*.*

It is well-known that a decidable parameterized problem is fixed-parameter tractable if and only if it admits a kernelization (see [7]). Most of the kernelization procedures are based on *reduction rules*. Next we define the notion of a *reduction rule* and when it is said to be *safe*.

Definition 2.3 (Safety of a Reduction Rule)**.** *A* reduction rule *that replaces an instance (I, k) of a parameterized problem Π by a reduced instance (I', k') is said to be* safe *when $(I, k) \in \Pi$ if and only if $(I', k') \in \Pi$.*

Tools and Techniques: One of the tools in our kernel result, and also in most of the polynomial kernel results mentioned in the related work section, uses the idea of *blocking sets* inspired from Jansen and Bodlaender [23]. A set X of vertices in $G \setminus S$ (where S is the modulator) is a blocking set in $G \setminus S$ if $\mathsf{vc}(G \setminus (S \cup X)) + |X| > \mathsf{vc}(G \setminus S)$. I.e. picking X into the solution results in an increase in the size of the vertex cover we get (compared to a minimum vertex cover). We show that the minimal blocking set of a connected component is bounded by a constant. Then using the idea of *chunks* (also introduced by Jansen and Bodlaender) and *conflicts*, we bound the number of connected components in $G \setminus S$ as a function of the size of the modulator. This bounds the size of the feedback vertex set in the graph by a function of k, as each component is a path or a cycle (and one vertex from each cycle component forms a feedback vertex set). Now we can use use the following theorem by Jansen and Bodlaender [23] to get a polynomial kernel.

Theorem 2.1. *(see [23])* Vertex Cover *parameterized by the size k, of a given feedback vertex set, has a kernel with $\mathcal{O}(k^3)$ vertices.*

In order to give a tighter upper bound for the kernel size, we exploit more structural properties of the modulator and use the following variation of Hall's theorem known as *expansion lemma* in some of the reduction rules.

Lemma 2.1 (q-Expansion Lemma)**.** (see [15, 32, 35]) *Let q be a positive integer and G be a bipartite graph with vertex partition A and B such that $|B| \geq q|A|$ and there are no isolated vertices in B. Then, there exists nonempty subsets $X \subseteq A, Y \subseteq B$ obtainable in polynomial time, such that*

- *there is a q-expansion of X into Y. I.e. there is a $M \subseteq E$ such that every vertex in X is incident with exactly q edges of M. Moreover M saturates exactly $q|X|$ vertices in Y, and*
- *$N_G(Y) \subseteq X$.*

3 Kernelization for VC-2-Mod

Throughout this section for an input (G, S, ℓ) to VC-2-Mod we use F to denote $V(G) \setminus S$. We use $|S| \leq k$ in Sections 3, 4, 5 and 6. Now, we are ready to describe the reduction rules that compress $G[F]$ to an equivalent instance whose size is polynomial in k. The rules will be applied sequentially. Note that, once a rule is applied, it is possible that some of the earlier rules may become applicable. Hence after every rule is applied, we start from the beginning and apply the first applicable reduction rule. We allow the input graph to have self loops, but no parallel edges. In particular, even if the input graph has no self loop, some reduction rules (for example, Reduction Rule 3.7) may create self loops. For every reduction rule, we denote the input instance as (G, S, ℓ) and the reduced instance as (G', S', ℓ'). We use $\leftarrow$ in the reduction rule to mean that the variable on the left is assigned the value in the right of the arrow.

3.1 Ensuring minimum degree three

The following reduction rules are standard for the Vertex Cover problem (see, for example, Chapter 4 of Downey and Fellows [11] for correctness of these rules).

Reduction Rule 3.1 (Isolated Vertex Rule)**.** *Remove isolated vertices from G. Formally, if u is an isolated vertex in G, then we set $G' \leftarrow G \setminus \{u\}, S' \leftarrow S \setminus \{u\}$ and $\ell' \leftarrow \ell$.*

Reduction Rule 3.2 (Self Loop Rule)**.** *If there exists a vertex $u \in V(G)$ such that there is a self loop with u, then we set $G' \leftarrow G \setminus \{u\}, S' \leftarrow S \setminus \{u\}$ and $\ell' \leftarrow \ell - 1$.*

Reduction Rule 3.3 (Degree 1 Rule)**.** *If there exists a vertex $u \in V(G)$ such that $deg_G(u) = 1$ and v is its unique neighbor, then we set $G' \leftarrow G \setminus \{u, v\}, S' \leftarrow S \setminus \{u, v\}$ and $\ell' \leftarrow \ell - 1$.*

Reduction Rule 3.4 (Degree 2 Rule)**.** *If there exists a vertex $u \in V(G)$ such that $deg_G(u) = 2$, let v, w be its two neighbors in G, then do the following:*

- *If $(v, w) \in E(G)$, then we set $G' \leftarrow G \setminus \{u, v, w\}, S' \leftarrow S \setminus \{u, v, w\}$ and $\ell' \leftarrow \ell - 2$.*
- *If $(v, w) \notin E(G)$, then set $G' \leftarrow (G \setminus \{u, v, w\}) \cup \{u_{new}\}, \ell' \leftarrow \ell - 1$, and make u_{new} adjacent to every vertex in $N_G(\{v, w\}) \setminus \{u\}$. If $\{u, v, w\} \cap S \neq \emptyset$, then we set $S' \leftarrow (S \setminus \{u, v, w\}) \cup \{u_{new}\}$. Otherwise we have that $\{u, v, w\} \cap S = \emptyset$ and we set $S' \leftarrow S$.*

Note that Reduction Rule 3.4 actually contracts the edges incident on u one by one. It is not immediate whether an application of Reduction Rule 3.4 would not increase the parameter. The following observation explains why it does not increase.

Observation 1. *Applications of Reduction Rules 3.1, 3.2, 3.3 and 3.4 do not increase the parameter for* VC-2-Mod *problem.*

Proof. Reduction Rules 3.1, 3.2, 3.3 only delete vertices and so do not increase the number of vertices in S. In Reduction Rule 3.4, for the first case, it is easy to see that the parameter does not increase as the vertices are only deleted. In the second case, when u, v, w are deleted from G and u_{new} has been added to G, if at least one of u, v, w was in S (i.e. $\{u, v, w\} \cap S \neq \emptyset$) before applying this rule, then the parameter does not increase despite the fact that u_{new} is added to S. Because at least one vertex is deleted from S and at most one vertex is added into S in such case. Let us consider the case when $u, v, w \in V(F)$ before applying this rule. Note that $N(u_{new}) = (N_G(v) \cup N_G(w)) \setminus \{u\}$ and v and w each has at most one neighbor other than u in F. In that case, u_{new} is added to F and this vertex has at most two neighbors in F. So, the parameter does not increase. $\square$

It is easy to see that Reduction Rules 3.1, 3.2, 3.3, and 3.4 can be performed in polynomial time. When the above reduction rules are not applicable, the minimum degree of G is 3. Therefore, every vertex $v \in F$ has at least one neighbor in S. Now we partition F into F_0, F_1 and F_2 such that every connected component of $G[F_0]$ is an isolated vertex, every connected component of $G[F_1]$ is either a path or a cycle of even length (length at least 4) and every connected component of $G[F_2]$ is an odd cycle. In particular we will have following notations.

- $F := V(G) \setminus S$.
- F_0 – every connected component of $G[F_0]$ is an isolated vertex.
- F_1 – every connected component of $G[F_1]$ is either a path or a cycle of even length.
- F_2 – every connected component of $G[F_2]$ is an odd cycle.

3.2 Reduction Rules to bound $G[F_0 \cup F_1]$

Our first step is to devise some reduction rules to bound the number of vertices in $G[F_0]$ and $G[F_1]$. In order to define such rules, we introduce a notion called Chunk. This notion will be useful not just for VC-2-MOD, but also for VC-PARAM-d-CVD. A variation of this definition was also given by Fomin and Strømme [16] and by Bodlaender and Jansen [23].

Definition 3.1 (Chunk). *Let (G, S, ℓ) be a* VC-2-MOD *(or* VC-PARAM-d-VCD*) instance. Then $X \subseteq S$ is a* Chunk *if the following properties hold.*

- $1 \leq |X| \leq 3$ *for* VC-2-MOD *instance.*
- $1 \leq |X| \leq d$ *for* VC-PARAM-d-CVD *instance.*
- *X is an independent set in G.*
- $|N_G(X) \cap F| + \mathsf{vc}(G[F \setminus N_G(X)]) \leq \mathsf{vc}(G[F]) + |S|$.

The intuition of defining Chunk is to find out suitable subsets of S which possibly could be *fully disjoint* from a minimum vertex cover. In other words, if a set $X \subseteq S$ *is not a* Chunk, that is, $|N_G(X) \cap F| + \mathsf{vc}(G[F \setminus N_G(X)]) \geq \mathsf{vc}(G[F]) + |S| + 1$ holds, then we will show that any minimum vertex cover of G must intersect X. A variation of the following lemma is due to Bodlaender and Jansen [23] (see also Fomin and Strømme [16]).

Lemma 3.1. *Let D be a minimum vertex cover of G. Then for all subsets $S' \subseteq S \setminus D$, $|N_G(S') \cap F| + \mathsf{vc}(G[F \setminus N_G(S')]) \leq \mathsf{vc}(G[F]) + |S|$.*

Proof. Let D be a minimum vertex cover of G and W be a minimum vertex cover of $G[F]$. That is, $|D| = \mathsf{vc}(G)$ and $|W| = \mathsf{vc}(G[F])$. Clearly $|D| \leq |S| + \mathsf{vc}(G[F])$ as $S \cup W$ forms a vertex cover of G.

For the sake of contradiction assume that there exists $S' \subseteq S \setminus D$ such that

$$|N_G(S') \cap F| + \mathsf{vc}(G[F \setminus N_G(S')]) \geq \mathsf{vc}(G[F]) + |S| + 1.$$

By the choice of S', we know that $S' \cap D = \emptyset$. Then, D must contain $N_G(S') \cap F$ from F and at least $\mathsf{vc}(G[F \setminus N_G(S')])$ vertices from F. So, $|D| \geq |N_G(S') \cap F| + \mathsf{vc}(G[F \setminus N_G(S')])$. By assumption $|N_G(S') \cap F| + \mathsf{vc}(G[F \setminus N_G(S')]) \geq \mathsf{vc}(G[F]) + |S| + 1$. Then $|D| \geq \mathsf{vc}(G[F]) + |S| + 1$, contradiction. $\square$

If D is a minimum vertex cover, then $X = S \setminus D$ is an independent set. Hence, we have the following corollaries.

Corollary 3.1. *Let (G, S, ℓ) be an instance of* VC-2-MOD *(or* VC-PARAM-d-CVD*) and let D be a minimum vertex cover of G. Then, any $X \subseteq S \setminus D$ where $1 \leq |X| \leq 3$ (or $1 \leq |X| \leq d$ respectively) is a* Chunk*.*

Corollary 3.2. *Let (G, S, ℓ) be an instance of* VC-2-MOD *(or* VC-PARAM-d-CVD*) and let D be a minimum vertex cover of G. Then, for any independent set $X \subseteq S$, where $1 \leq |X| \leq 3$ (or $1 \leq |X| \leq d$ respectively), if X is not a* Chunk*, then $D \cap X \neq \emptyset$.*

Now we have the following Reduction Rule.

Reduction Rule 3.5 (Non-Chunk-Rule-I). *If there exists a vertex $x \in S$ such that $\{x\}$ is not a* Chunk *(or formally $\mathsf{vc}(G[F \setminus N_G(x)]) + |N_G(x) \cap F| \geq \mathsf{vc}(G[F]) + |S| + 1$), then we set $G' \leftarrow G \setminus \{x\}$, $S' \leftarrow S \setminus \{x\}$ and $\ell' \leftarrow \ell - 1$.*

The proof of the next result follows from Corollary 3.2.

Lemma 3.2. *Reduction Rule 3.5 is safe.*

Next we generalize Non-Chunk-Rule-I to non-Chunks of size 2.

Reduction Rule 3.6 (Non-Chunk-Rule-II)**.** *If there exists $x, y \in S, (x, y) \notin E(G)$ such that $\{x, y\}$ is not a* Chunk *(or formally $\mathsf{vc}(G[F \setminus N_G(\{x, y\})]) + |N_G(\{x, y\}) \cap F| \geq \mathsf{vc}(G[F]) + |S| + 1$), then add the edge (x, y) to get the new graph G'. We set $S' \leftarrow S$ and $\ell' \leftarrow \ell$.*

Lemma 3.3. *Reduction Rule 3.6 is safe.*

Proof. The proof of this lemma is also based on the Corollary 3.2. Suppose D is a minimum vertex cover of G. By Corollary 3.2 we know that $D \cap X \neq \emptyset$. By the precondition of the reduction rule, $\{x, y\}$ is not a Chunk. So, any minimum vertex cover of G must contain at least one of x or y. Adding an edge between x and y captures this constraint. Hence the reduction rule is safe. □

Remark 1. *Let $\mathcal{F}$ be a graph class that is hereditary and closed under contraction of edges. Furthermore, suppose that* Vertex Cover *is polynomial time solvable in any graph belonging to graph class $\mathcal{F}$. Let S be a modulator to a graph $F \in \mathcal{F}$. Then, we note that the proofs of correctness of the Reduction Rules 3.1 to 3.6 also apply when the parameter is $|S|$, i.e.*

- *Reduction Rules 3.1 to 3.6 are applicable to* Vertex Cover *parameterized by $|S|$, and*
- *Reduction Rules 3.1 to 3.6 do not increase $|S|$.*

Note that while Reduction Rule 3.6 does not decrease the size of the graph or ℓ, it does enable the applicability of some rules (for example, Reduction Rule 3.7). Our next reduction rule is a slight variation of the one proposed by Jansen and Bodlaender [23] for Vertex Cover parameterized by the feedback vertex set number.

Reduction Rule 3.7 (Edge Rule)**.** *If there exists an edge $(u, v) \in E(G[F])$ such that $(N_G(u) \cap S) \cap (N_G(v) \cap S) = \emptyset$ and for all $x \in N_G(u) \cap S$, for all $y \in N_G(v) \cap S$ we have that $(x, y) \in E(G)$, then do the following.*

- *Delete u, v from the graph.*
- *If u has a neighbor t in F that is not v, then make t adjacent to every vertex in $N_G(v) \cap S$.*
- *If v has a neighbor w in F that is not u, then make w adjacent to every vertex in $N_G(u) \cap S$.*
- *If the vertices t, w exist, then they are unique and add the edge (t, w). Note that if $t \neq w$ and (t, w) was already an edge in G, then we don't add any extra edge (t, w). If $t = w$ (this case can happen when $(u, v) \in E(G[F_2])$), then we add a self loop in t.*
- *Set ℓ' to $\ell - 1$.*

Let G' be the new graph and we set $S' \leftarrow S$ (see Figure 2 for an illustration).

Lemma 3.4. *Reduction Rule 3.7 is safe.*

Proof. We give the proof for different cases as vertices t and w both need not necessarily exist.

1. **Both t and w exist:** Let G' be the graph obtained after applying the reduction rule on G. Observe that since for all $x \in N_G(u) \cap S$ and for all $y \in N_G(v) \cap S$, we have that $(x, y) \in E(G)$, any vertex cover (not necessary minimum) of G (and also G') must contain *either $N_G(u) \cap S$ or $N_G(v) \cap S$ (or both).*

   We first give the proof for the reverse direction of the reduction rule. Let D be a vertex cover for G' of size at most $\ell - 1$. Recall, that $(t, w) \in E(G')$. Now based on which vertice(s) from $\{t, w\}$ belong to D, we have the following cases.

   - Suppose that $\mathbf{t \in D, w \notin D}$. By our construction, we made w adjacent to every vertex in $N_G(u) \cap S$ in G'. Thus, $N_G(u) \cap S \subseteq D$. Given this observation one can easily show that $D \cup \{v\}$ is a vertex cover for G of size at most ℓ.

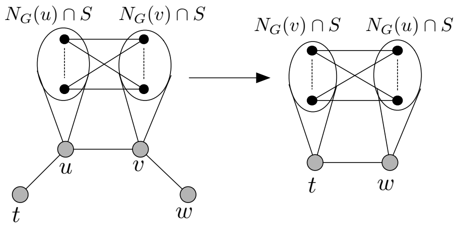

Figure 2: An illustration of Reduction Rule 3.7.

- Suppose that $\mathbf{t} \notin \mathbf{D}, \mathbf{w} \in \mathbf{D}$. By our construction, we made t adjacent to every vertex in $N_G(v) \cap S$ in G'. Thus, $N_G(v) \cap S \subseteq D$. Given this observation, as before, one can easily show that $D \cup \{u\}$ is a vertex cover for G of size at most ℓ.
- Suppose that $\mathbf{t}, \mathbf{w} \in \mathbf{D}$. We know that in both G as well as in G', either $N_G(u) \cap S \subseteq D$ or $N_G(v) \cap S \subseteq D$. If $N_G(u) \cap S \subseteq D$, then $D \cup \{v\}$ is a vertex cover for G of size at most ℓ, else $D \cup \{u\}$ is a vertex cover for G of size at most ℓ.

We now give the proof for the forward direction of the reduction rule. Let D be a vertex cover of G of size at most ℓ. We have several cases based on which endpoint(s) of the edge $(u, v) \in E(G)$ belong to D.

- Suppose that $\mathbf{u} \in \mathbf{D}, \mathbf{v} \notin \mathbf{D}$. Since $v \notin D$, we have that $N_G(v) \subseteq D$. This also implies that $w \in D$. Let $D' = D \setminus \{u\}$. We know that the edges of the form (w, x) are added in G' where $x \in N_G(u) \cap S$. D' overs these edges because $w \in D'$. Also the edges of the form (t, y) are added in G' where $y \in N_G(v) \cap S$. As $N_G(v) \cap S \subseteq D$, hence $N_G(v) \cap S \subseteq D'$. Hence these edges are also covered by D'. Also the edge (t, w) is added if it was not an edge in G. As $w \in D'$, this edge is also covered by D'. Thus, D' is a vertex cover of G' of size at most $\ell - 1$.
- Suppose that $\mathbf{u} \notin \mathbf{D}, \mathbf{v} \in \mathbf{D}$. By our assumption, we have that $N_G(u) \subseteq D$. This also implies that $t \in D$. Let $D' = D \setminus \{v\}$. By an argument similar to the previous case we can prove that D' is a vertex cover of G' of size at most $\ell - 1$.
- Suppose that $\mathbf{u}, \mathbf{v} \in \mathbf{D}$. By the precondition of the reduction rule, either $N_G(u) \cap S \subseteq D$ or $N_G(v) \cap S \subseteq D$. Suppose $N_G(u) \cap S \subseteq D$. Then let $D' = (D \setminus \{u, v\}) \cup \{t\}$. We know that the edges of the form (w, x) are added in G' where $x \in N_G(u) \cap S$. D' covers these edges because $N_G(u) \cap S \subseteq D'$. Also the edges of the form (t, y) are added in G' where $y \in N_G(v) \cap S$. As $t \in D'$, these edges are also covered by D'. Also the edge (t, w) is added if it was not an edge in G. This edge is also covered by t. So, when $N_G(u) \cap S \subseteq D$, then $(D \setminus \{u, v\}) \cup \{t\}$ is a vertex cover of G' of size at most $\ell - 1$. Similarly if $N_G(v) \cap S \subseteq D$, then $(D \setminus \{u, v\}) \cup \{w\}$ is a vertex cover of G' of size at most $\ell - 1$.

Note that when $t = w$, then the component in which this reduction rule is applied is a triangle. Also an edge is added between t and any vertex $a \in (N_G(u) \cup N_G(v)) \cap S$. A self loop gets created in vertex t as well. In that case for reverse direction, we have that t must be a part of any solution. We have already mentioned that after application of this reduction rule, we again check if some rule from Reduction Rules 3.1, 3.2, 3.3, 3.4, 3.5, 3.6 is applicable. In this case when $t = w$, then Reduction Rule 3.2 becomes applicable and hence vertex t also gets deleted immediately after that.

2. **Vertex t exists but w does not exist:**
($\Leftarrow$) Let D be a vertex cover of G'.

- If $t \in D$. There are two cases now. We know that either $N_G(u) \cap S \subseteq D$ or $N_G(v) \cap S \subseteq D$. If $N_G(u) \cap S \subseteq D$, then $D \cup \{v\}$ is a vertex cover of G. Symmetrically if $N_G(v) \cap S \subseteq D$, then $D \cup \{u\}$ is a vertex cover of G.
- If $t \notin D$, then $N_{G'}(t) \cap S \subseteq D$. In that case, $N_G(v) \cap S \subseteq D$. So, $D \cup \{u\}$ is a vertex cover of G.

($\Rightarrow$) Let D be a vertex cover of G.

- Suppose that $u \in D, v \notin D$. Then $N_G(v) \cap S \subseteq D$. Now note that the edges of the form (t, x) are added in G' where $x \in N_G(v) \cap S$. Consider $D' = D \setminus \{u\}$. As $N_G(v) \cap S \subseteq D'$, the newly added edges are also covered. So D' is a vertex cover of G' of size at most $\ell - 1$.
- Suppose that $u \notin D, v \in D$. Then $N_G(u) \cap S \subseteq D$. Also $t \in D$. Consider $D' = D \setminus \{v\}$. Now note that the edges of the form (t, x) are added in G' where $x \in N_G(v) \cap S$. As $t \in D'$, such edges are covered by D'. So D' is a vertex cover of G' of size at most $\ell - 1$.
- Suppose that $u, v \in D$. Even in this case also either $N_G(u) \cap S \subseteq D$ or $N_G(v) \cap S \subseteq D$. If $N_G(u) \cap S \subseteq D$, then we say that $(D \setminus \{u, v\}) \cup \{t\}$ is a vertex cover of G' as the newly added edges are (t, x) with $x \in N_G(v) \cap S$. Otherwise if $N_G(v) \cap S \subseteq D$, then $D \setminus \{u, v\}$ is a vertex cover of G' as the newly added edges are being covered by $N_G(v) \cap S$ (if not by t already).

3. **Vertex w exists but t does not exist:** This case is symmetric to Case 2 above.
4. **Neither w nor t exists:** In this case, we just delete u and v from G. As $N_G(u) \cap S$ and $N_G(v) \cap S$ form complete bipartite graph, we can argue easily that exactly one from u and v is part of any optimal solution. Note that $\mathsf{vc}(G) \geq \mathsf{vc}(G') + 1$ as $(u, v) \in E(G)$. So the forward direction easily follows. Now let D' be a vertex cover of G'. Then if $N_G(u) \cap S \subseteq D'$, we construct $D = D' \cup \{v\}$. Clearly D covers the edge (u, v) and all other edges incident to u are already covered. Hence D is a vertex cover of G. Similarly if $N_G(v) \cap S \subseteq D'$, we construct $D = D' \cup \{u\}$ which is a vertex cover of G. So $\mathsf{vc}(G) \leq \mathsf{vc}(G') + 1$. So the reduction rule is safe.

This completes the proof the lemma. ☐

Lemma 3.5. *Reduction Rules 3.5, 3.6, and 3.7 can be implemented in polynomial time.*

Proof. Reduction Rule 3.5 requires us to check if the condition $\mathsf{vc}(G[F \setminus N_G(x)]) + |N_G(x) \cap F| \geq \mathsf{vc}(G[F]) + |S| + 1$ holds for a vertex in S. Since $G[F]$ is a graph of degree at most two, minimum vertex cover of $G[F]$ can be computed in polynomial time. $G[F \setminus N_G(x)]$ is an induced subgraph of $G[F]$. Therefore, minimum vertex cover of $G[F \setminus N_G(x)]$ also can be computed in polynomial time. Thus, Reduction Rule 3.5 can be performed in polynomial time. By a similar argument as above, we can show that Reduction Rule 3.6 can be applied in polynomial time.

To apply Reduction Rule 3.7, we need to check if there exists an edge $(u, v) \in E(G[F])$ such that $N_G(u) \cap N_G(v) \cap S = \emptyset$. Suppose $N_G(u) \cap N_G(v) \cap S = \emptyset$, then we check whether for every $x \in N_G(u) \cap S$ and for every $y \in N_G(v) \cap S$, $(x, y) \in E(G)$. If we succeed here also then we need to modify G locally to obtain G'. Clearly, all this can be done in polynomial time. This completes the proof. ☐

Now, in order to bound the number of vertices in F_0 and F_1, we proceed to summarize some of the properties of $G[F_0 \cup F_1]$ when Reduction Rules 3.1 to 3.7 are no longer applicable. If $X \subseteq V(G)$, then clearly $\mathsf{vc}(G) \leq |X| + \mathsf{vc}(G \setminus X)$. The following lemma shows that when X satisfies some interesting properties, then we can prove a stronger inequality.

Lemma 3.6. *Let $G = (V, E)$ be a graph. Then the following statements are true.*

1. *Let V_0 be the set of isolated vertices in G. Let $X \subseteq V(G)$ such that $|X \cap V_0| = r$, then $|X| + \mathsf{vc}(G \setminus X) \geq \mathsf{vc}(G) + r$.*
2. *Let G be a bipartite graph and M be a maximum matching in G. If $X \subseteq V(G)$ be such that X contains both the endpoints of r edges of M, then $|X| + \mathsf{vc}(G \setminus X) \geq \mathsf{vc}(G) + r$.*

Proof. Let $G = (V, E)$ be a graph. We prove the statements in the given order.

1. Consider any minimum vertex cover D^* of G. As vertices of V_0 do not cover any edge, $D^* \cap V_0 = \emptyset$. So, $|D^*| = \mathsf{vc}(G) = \mathsf{vc}(G \setminus V_0)$. Clearly,

$$|X| + \mathsf{vc}(G \setminus X) = |X \cap V_0| + |X \setminus V_0| + \mathsf{vc}(G \setminus X)$$

and $|X \setminus V_0| + \mathsf{vc}(G \setminus X) \geq \mathsf{vc}(G \setminus V_0) = \mathsf{vc}(G)$. So we get the following

$$|X \cap V_0| + |X \setminus V_0| + \mathsf{vc}(G \setminus X) \geq r + \mathsf{vc}(G)$$

$$|X| + \mathsf{vc}(G \setminus X) \geq \mathsf{vc}(G) + r$$

2. Let G be a bipartite graph and M be a maximum matching in G. Given $X \subseteq V(G)$, let X contains both endpoints of r edges of M. We know by König's Theorem (see [10]) that $\mathsf{vc}(G) = |M|$.

Let $M' = \{(u, v) \in M | u, v \in X\}$ be the set of r edges of M with both endpoints in X.

$$|X| + \mathsf{vc}(G \setminus X) = |V(M') \cap X| + |X \setminus V(M')| + \mathsf{vc}(G \setminus X) = 2r + |X \setminus V(M')| + \mathsf{vc}(G \setminus X)$$

Now we know that $|X \setminus V(M')| + \mathsf{vc}(G \setminus X) \geq \mathsf{vc}(G \setminus V(M'))$. Now $\mathsf{vc}(G \setminus V(M'))$ also has a matching of size $|M| - |M'| = |M| - r$. So, $\mathsf{vc}(G \setminus V(M')) \geq |M| - r$. So we have that

$$2r + |X \setminus V(M')| + \mathsf{vc}(G \setminus X) \geq 2r + \mathsf{vc}(G \setminus V(M')) \geq 2r + |M| - r = \mathsf{vc}(G) + r$$

This completes the proof of the lemma. □

Using the above lemma, we get the following properties of $G[F_0 \cup F_1]$ when Reduction Rules 3.1 to 3.7 are not applicable.

Lemma 3.7. *Let (G, S, ℓ) be an instance to* VC-2-MOD *on which Reduction Rules 3.1 to 3.7 are not applicable. Furthermore, let F, F_0, F_1 and F_2 be as defined previously and let M be a maximum matching of $G[F_1]$. Then the following statements are true.*

1. *For every $x \in S$, $|N_G(x) \cap F_0| \leq |S|$.*
2. *For every $x \in S$, $N_G(x) \cap F_1$ contains both the endpoints of at most $|S|$ edges of M.*
3. *For every $(x, y) \notin E(G[S])$, $N_G(\{x, y\}) \cap F_1$ contains both the endpoints of at most $|S|$ edges of M.*
4. *For every edge $(u, v) \in E(G[F])$, either $N_G(u) \cap N_G(v) \cap S \neq \emptyset$ or there exist $s \in N_G(u) \cap S, t \in N_G(v) \cap S$ such that $(s, t) \notin E(G)$.*

Proof. We prove the statements in the given order.

1. Suppose that there exists a vertex $x \in S$, such that $|N_G(x) \cap F_0| = r \geq |S| + 1$. Let $X = N_G(x) \cap F$. Then $|X \cap F_0| = r \geq |S| + 1$. Hence by Lemma 3.6, we have that $|X| + \mathsf{vc}(G[F \setminus X]) \geq \mathsf{vc}(G[F]) + r \geq \mathsf{vc}(G[F]) + |S| + 1$. Then, $\{x\}$ is not a Chunk. Therefore, Reduction Rule 3.5 becomes applicable which is a contradiction. Hence, $|N_G(x) \cap F_0| \leq |S|$.
2. Suppose the claim is not true. Let $N_G(x)$ contains both the the endpoints of r edges of M. By our assumption, $r \geq |S| + 1$. Let $X = N_G(x) \cap F$. Now consider $\mathsf{vc}(G[F \setminus X]) + |X|$. We know the following.

$$\mathsf{vc}(G[F \setminus X]) + |X| = \mathsf{vc}(G[(F_1 \cup F_0) \setminus X]) + \mathsf{vc}(G[F_2 \setminus X]) + |X \cap (F_1 \cup F_0)| + |X \cap F_2|$$

Now, we have that $\mathsf{vc}(G[F_2 \setminus X]) + |X \cap F_2| \geq \mathsf{vc}(G[F_2])$. We know that $G[F_0 \cup F_1]$ is bipartite. Recall that $X = N_G(x) \cap F$ and X contains both endpoints of r edges of M. Also M is a maximum matching in $G[F_0 \cup F_1]$. Hence, in particular $X \cap (F_0 \cup F_1)$ also contains both endpoints of r edges of M. By Lemma 3.6, we have that $\mathsf{vc}(G[(F_1 \cup F_0) \setminus X]) + |X \cap (F_1 \cup F_0)| \geq \mathsf{vc}(G[F_1 \cup F_0]) + r \geq \mathsf{vc}(G[F_1 \cup F_0]) + |S| + 1$. Then,

$$\begin{aligned}\mathsf{vc}(G[F \setminus X]) + |X| &= \mathsf{vc}(G[(F_1 \cup F_0) \setminus X]) + \mathsf{vc}(G[F_2 \setminus X]) + |X \cap (F_1 \cup F_0)| + |X \cap F_2| \\ &\geq \mathsf{vc}(G[F_0 \cup F_1]) + |S| + 1 + \mathsf{vc}(G[F_2]) = \mathsf{vc}(G[F]) + |S| + 1\end{aligned}$$

Then, $\{x\}$ is not a Chunk. But then Reduction Rule 3.5 becomes applicable which is a contradiction.

3. Suppose the claim is not true. Let (x, y) be a pair of nonadjacent vertices of S such that $X = N_G(\{x, y\}) \cap F$ that contains both the endpoints of r edges of M. By our assumption, $r \geq |S| + 1$. Now consider $\mathsf{vc}(G[F \setminus X]) + |X|$. We know that

$$\mathsf{vc}(G[F \setminus X]) + |X| = \mathsf{vc}(G[(F_1 \cup F_0) \setminus X]) + \mathsf{vc}(G[F_2 \setminus X]) + |X \cap (F_1 \cup F_0)| + |X \cap F_2|$$

Now, we have that $\mathsf{vc}(G[F_2 \setminus X]) + |X \cap F_2| \geq \mathsf{vc}(G[F_2])$. We know that $G[F_0 \cup F_1]$ is bipartite. Recall that $X = N_G(\{x, y\}) \cap F$ and X contains both endpoints of r edges of M. Also M is a maximum matching in $G[F_0 \cup F_1]$. Hence, in particular $X \cap (F_0 \cup F_1)$ also contains both endpoints of r edges of M. By Lemma 3.6, we have that $\mathsf{vc}(G[(F_1 \cup F_0) \setminus X]) + |X \cap (F_1 \cup F_0)| \geq \mathsf{vc}(G[F_1 \cup F_0]) + r \geq \mathsf{vc}(G[F_1 \cup F_0]) + |S| + 1$. Then,

$$\begin{aligned}\mathsf{vc}(G[F \setminus X]) + |X| &= \mathsf{vc}(G[(F_1 \cup F_0) \setminus X]) + \mathsf{vc}(G[F_2 \setminus X]) + |X \cap (F_1 \cup F_0)| + |X \cap F_2| \\ &\geq \mathsf{vc}(G[F_0 \cup F_1]) + |S| + 1 + \mathsf{vc}(G[F_2]) = \mathsf{vc}(G[F]) + |S| + 1\end{aligned}$$

Then, $\{x, y\}$ is not a Chunk. But then Reduction Rule 3.6 becomes applicable which is a contradiction.

4. As Reduction Rules 3.1, 3.2, 3.3 and 3.4 are not applicable, every vertex in F has at least one neighbor in S. So, for any $u \in F$, we have that $N_G(u) \cap S \neq \emptyset$. If there exists an edge $(u, v) \in E(G[F])$ such that $N_G(u) \cap N_G(v) \cap S = \emptyset$ and for every $x \in N_G(u) \cap S$, for every $y \in N_G(v) \cap S$ we have that $(x, y) \in E(G)$, then Reduction Rule 3.7 is applicable. So, its contrapositive statement would be the following. If Reduction Rule 3.7 is not applicable, then for any edge $(u, v) \in E(G[F])$ either $N_G(u) \cap N_G(v) \cap S \neq \emptyset$ or there exist $s \in N_G(u) \cap S$ and $t \in N_G(v) \cap S$ such that $(s, t) \notin E(G)$.

This completes the proof the lemma. □

Now, we are ready to bound the number of vertices in $G[F_1 \cup F_0]$ to show the following.

Lemma 3.8. *When Reduction Rules 3.1 to 3.7 are not applicable then $G[F_1 \cup F_0]$ has $\mathcal{O}(k^3)$ vertices.*

Proof. By inapplicability of Reduction Rules 3.1, 3.2, 3.3, 3.4, we know that every vertex in F (in fact $F_0 \cup F_1$) has at least one neighbor in S. From Lemma 3.7, we see that for every $x \in S$, $|N_G(x) \cap F_0| \leq |S|$. Therefore, $|F_0| \leq |S|^2 = k^2$.

We define a function $f : E(G[F_1]) \mapsto S \cup \binom{S}{2}$ as follows.

$$f((u, v)) = \begin{cases} x & \text{if } x \in N_G(u) \cap N_G(v) \cap S \text{ (fix an } x \text{ arbitrarily)} \\ (x, y) & \text{if there exist } x \in N_G(u) \cap S, y \in N_G(v) \cap S \text{ s.t. } (x, y) \notin E(G) \text{ (fix an } (x, y) \text{ arbitrarily)} \end{cases}$$

We know that for every edge $(u, v) \in G[F_1]$, either an $x \in S$ or a pair $(x, y) \in \binom{S}{2}$ as defined above, exists as Reduction Rules 3.4 and 3.7 are not applicable. So, f is well defined. For every edge $(u, v) \in E(G[F])$ we associate $x \in S$ (or $(x, y) \in \binom{S}{2}$) as provided by function f. Fix a maximum matching M from $G[F_1]$. Now for every $x \in S$ (and also for every $(x, y) \in \binom{S}{2}$ such that $(x, y) \notin E(G)$), $N_G(x) \cap F_1$ (and also $N_G(\{x, y\}) \cap F_1$)) contains both the endpoints of at most $|S|$ edges in M by Lemma 3.7. So a vertex in S or a non-edge between a pair of points in S is a pre-image of f of at most $|S|$ edges of $E(G[F_1]) \cap M$. Therefore

$$|E(G[F_1]) \cap M| \leq (|S| + \binom{|S|}{2})|S| \leq (k + \binom{k}{2})k$$

So the number of edges in F_1 matched by M is $\mathcal{O}(k^3)$. The number of vertices in F_1 that are not matched by M are at most $|M|$. Therefore, $|F_1|$ is $\mathcal{O}(k^3)$. So, $G[F_0 \cup F_1]$ has $\mathcal{O}(k^3)$ vertices. □

Note that though the Reduction Rule 3.7 is applicable for any edge in $G[F]$, in Lemma 3.8, we apply it only for the edges in $G[F_1]$ and hence we cannot yet deduce a bound on the total number of edges in $G[F]$ (in particular in $G[F_2]$). This is because the proof of Lemma 3.7 uses the fact that M is a matching in the bipartite graph $G[F_1]$. However, in Sections 3.3 and 3.4 we will develop a few more reduction rules that can help us to bound the number of edges in $G[F_2]$.

3.3 Bounding the number of odd cycles

Now to obtain the kernel, all we are left to do is to bound the size of $G[F_2]$. Here we first give rules to bound the *number* of odd cycles. As every component in $G[F_2]$ is an odd cycle, we interchangeably use the term component or odd cycle to mean the same thing in $G[F_2]$. Central to the rules in this subsection is the notion of a *blocking set*, we define the notion and prove some properties about them first.

Definition 3.2 (Blocking Set and Good Set)**.** *Let $B \subseteq V(G)$. We call B to be a blocking set if $\mathsf{vc}(G[V(G) \setminus B]) + |B| > \mathsf{vc}(G)$. We call a blocking set B to be a minimal blocking set if no proper subset of B is a blocking set. A set $B \subseteq V(G)$ is called a good set if it is not a blocking set.*

If an algorithm picks the vertices of a blocking set B into a solution, then any way to complete it to a feasible vertex cover results in a non-optimal solution. So the blocking set does not allow us to complete it to an optimal solution and forces to produce a sub-optimal solution. For example, in a cycle of even length, $\{u, v\}$ is a blocking set for any edge $e = (u, v)$ as no optimum solution for the cycle contains two vertices of the same edge. We will *apply these notions* to $G[F]$. Based on Reduction Rules 3.5 and 3.6, we can prove the following lemma.

Lemma 3.9. *When Reduction Rules 3.5 and 3.6 are not applicable, the following statements are true.*

1. *For every $x \in S$, $N_G(x) \cap F$ contains a blocking set in at most $|S|$ cycles in $G[F_2]$.*
2. *For every $x, y \in S, (x, y) \notin E(G)$, $N_G(\{x, y\}) \cap F$ contains a blocking set in at most $|S|$ cycles in $G[F_2]$.*

Proof. The proof of this lemma uses arguments that are similar to the arguments in Lemma 3.7 and some properties of blocking set in odd cycles. We prove the statements in the given order.

1. Suppose that $N_G(x) \cap F$ contains a blocking set in at least $|S| + 1$ odd cycles. Then there are at least $|S| + 1$ cycles $D_1, D_2, \ldots, D_{|S|+1} \in G[F_2]$ such that for each $j \in [|S| + 1]$, $|N_G(x) \cap D_j| + \mathsf{vc}(D_j \setminus N_G(x)) \geq \mathsf{vc}(D_j) + 1$. So, $|N_G(x) \cap F_2| + \mathsf{vc}(G[F_2 \setminus N_G(x)]) \geq \mathsf{vc}(G[F_2]) + |S| + 1$. So,
$$|N_G(x) \cap F| + \mathsf{vc}(G[F \setminus N_G(x)])$$
$$= |N_G(x) \cap F_2| + \mathsf{vc}(G[F_2 \setminus N_G(x)]) + |N_G(x) \cap (F_1 \cup F_0)| + \mathsf{vc}(G[(F_1 \cup F_0) \setminus N_G(x)])$$
$$\geq \mathsf{vc}(G[F_2]) + |S| + 1 + \mathsf{vc}(G[F_1 \cup F_0]) = \mathsf{vc}(G[F]) + |S| + 1$$
It means that $\{x\}$ is not a Chunk. So, Reduction Rule 3.5 is applicable which is a contradiction.
2. Suppose that $N_G(\{x, y\}) \cap F$ contains a blocking set in at least $|S| + 1$ odd cycles. Then there are at least $|S| + 1$ cycles $D_1, D_2, \ldots, D_{|S|+1} \in G[F_2]$ such that for each $j \in [|S| + 1]$, $|N_G(\{x, y\}) \cap D_j| + \mathsf{vc}(D_j \setminus N_G(\{x, y\})) \geq \mathsf{vc}(D_j) + 1$. So, $|N_G(\{x, y\}) \cap F_2| + \mathsf{vc}(G[F_2 \setminus N_G(\{x, y\})]) \geq \mathsf{vc}(G[F_2]) + |S| + 1$. So,
$$|N_G(\{x, y\}) \cap F| + \mathsf{vc}(G[F \setminus N_G(\{x, y\})])$$
$$= |N_G(\{x, y\}) \cap F_2| + \mathsf{vc}(G[F_2 \setminus N_G(\{x, y\})]) + |N_G(\{x, y\}) \cap (F_1 \cup F_0)| + \mathsf{vc}(G[(F_1 \cup F_0) \setminus N_G(\{x, y\})])$$
$$\geq \mathsf{vc}(G[F_2]) + |S| + 1 + \mathsf{vc}(G[F_1 \cup F_0]) = \mathsf{vc}(G[F]) + |S| + 1$$
So, $\{x, y\}$ is not a Chunk. It means that Reduction Rule 3.6 is applicable which is a contradiction.

This completes the proof of the lemma. $\square$

We now show that the above lemma already suffices to bound the number of some specific types of components in $G[F_2]$. Towards that we first partition the set of components in $G[F_2]$ as $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ and $\mathcal{B}_4$ as follows.

- $\mathcal{B}_1 = \{C | C$ is a component in $G[F_2]$ and there exists $x \in S$ such that $N_G(x) \cap C$ contains a blocking set in $C\}$.
- $\mathcal{B}_2 = \{C | C$ is a component in $G[F_2]$ and there exist $x, y \in S, (x, y) \notin E(G)$ such that $C \cap (N_G(x) \cup N_G(y))$ contains a blocking set in $C\} \setminus \mathcal{B}_1$.
- $\mathcal{B}_3 = \{C | C$ is a component in $G[F_2]$ and there exist $x, y, z \in S, \{x, y, z\}$ is independent set such that $C \cap (N_G(x) \cup N_G(y) \cup N_G(z))$ contains a blocking set in $C\} \setminus (\mathcal{B}_1 \cup \mathcal{B}_2)$.
- $\mathcal{B}_4 = \{$Components in $G[F_2]$ that are not in $\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{B}_3\}$.

Note that if $C \in \mathcal{B}_2$ then for any blocking set B in C, $B \not\subset N_G(x)$ for any $x \in S$ and if $C \in \mathcal{B}_3$, then for any blocking set B in C, $B \not\subset N_G(A)$ for any independent set A of size at most two in S.

The following corollaries follow immediately from Lemma 3.9.

Corollary 3.3. *When Reduction Rules 3.5 and 3.6 are not applicable,* $|\mathcal{B}_1| \leq k^2$.

Proof. We define a function $f : \mathcal{B}_1 \mapsto S$ such that $f(C) = x$ where $N_G(x) \cap C$ contains a blocking set in C (if there are more than one vertex, choose one arbitrarily).

We know that such an x exists for every $C \in \mathcal{B}_1$ by definition of $\mathcal{B}_1$. Now by Lemma 3.9, for every $x \in S$ there are at most $|S| \leq k$ cycles in $G[F_2]$ such that $N_G(x)$ contains a blocking set in each of those cycles. So for every $x \in S$ there are at most $|S| \leq k$ cycles in $\mathcal{B}_1$ such that $N_G(x)$ contains a blocking set in each of those cycles. Therefore, $|\mathcal{B}_1| \leq k^2$. $\square$

Corollary 3.4. *When Reduction Rules 3.5 and 3.6 are not applicable,* $|\mathcal{B}_2| \leq k\binom{k}{2}$.

Proof. We define a function $f : \mathcal{B}_2 \mapsto \binom{S}{2}$ such that $f(C) = (x, y)$ where $(x, y) \notin E(G), x, y \in S$ and $N_G(\{x, y\}) \cap C$ contains a blocking set in C (if there is more than one such pair, then choose one pair arbitrarily).

We know that such a nonadjacent pair (x, y) exists for every $C \in \mathcal{B}_2$ by definition of $\mathcal{B}_2$. Now by Lemma 3.9, for every $x, y \in S, (x, y) \notin E(G)$ there are at most $|S| \leq k$ cycles in $G[F_2]$ such that $N_G(\{x, y\})$ contains a blocking set in each of those cycles. So for every $x, y \in S, (x, y) \notin E(G)$ there are at most $|S| \leq k$ cycles in $\mathcal{B}_2$ such that $N_G(\{x, y\})$ contains a blocking set in each of those cycles. Therefore, $|\mathcal{B}_2| \leq k\binom{k}{2}$. $\square$

Now, in the rest of the section, we bound the number of remaining components, in particular those that have blocking sets that are neighbors to more than two vertices of S or those that have no blocking sets. Our main observation that helps to do this is that in an odd cycle, minimal blocking sets are of size exactly three.

Properties of Blocking Set: Let C be a cycle in $G[F_2]$ and let its vertices be ordered *clockwise* as $a_0, a_1, \ldots, a_{|C|-1}$. Let a_i, a_j be two vertices of C. Then by $dist(a_i, a_j)$, we mean the number of edges in the *clockwise path* in C that goes through the vertices $a_{i+1}, a_{i+2}, \ldots, a_{j-1}$ where the subscripts are taken mod $|C|$. The following statement is easy to verify.

Observation 2. *In a cycle, no single vertex forms a blocking set and hence minimal blocking sets are of size at least* 2.

It is easy to see that the following remark holds true for a minimal blocking set in $G[F]$.

Remark 2. *Let $B \subseteq F$ be a minimal blocking set in $G[F]$. Then there exists a unique C for which $B \subseteq V(C)$ where C is a component of $G[F]$.*

Theorem 3.1.
- *Let $C = \{a_1, \ldots, a_{2r+1}\}$ be an odd cycle and $B \subseteq V(C)$. If B is a minimal blocking set, then $|B| = 3$ and the clockwise distance between every pair of consecutive vertices in B is odd.*

- *Let $C = \{a_1, \ldots, a_{2r}\}$ be an even cycle and $B \subseteq V(C)$. If B is a minimal blocking set, then $|B| = 2$ and the clockwise distance between every pair of consecutive vertices in B is odd.*

Proof. To prove Theorem 3.1, we need to establish the following lemmas.

Lemma 3.10. *Let $C = \{a_1, \ldots, a_r\}$ be a cycle, and let $B \subseteq C$ of size q at least 2, such that at most one consecutive pair of vertices $a_i, a_{j \bmod r} \in B, i < j$ is such that $dist(a_i, a_{j \bmod r})$ is odd. Then B is a good set.*

Proof. Let $i_1 < i_2 < \ldots < i_q$ be the vertices of B in the order in which they appear in C. Suppose that $dist(a_{i_l \bmod r}, a_{i_{(l+1)} \bmod r})$ is even for every l. Then, each of the q components of $C \setminus B$ is a path with an odd number of vertices. Let p_i be the number of vertices of the i-th component (in some order), for $i = 1$ to q whose vertex cover number is $(p_i - 1)/2$. Thus we have, $\mathsf{vc}(C \setminus B) + |B| = \sum_{i=1}^{q} \frac{(p_i-1)}{2} + q = \sum_{i=1}^{q} \frac{(p_i+1)}{2}$. But $\sum_{i=1}^{q} p_i + q = |C| = r$ and hence $\mathsf{vc}(C \setminus B) + |B| = |C|/2 = r/2 \leq \mathsf{vc}(C)$. The last inequality follows as if r is even, then $\mathsf{vc}(C) = r/2$ and is $(r+1)/2$ otherwise. Hence B is a good set.

Suppose that for exactly one l, $dist(a_{i_l \bmod r}, a_{i_{l+1} \bmod r})$ is odd. Then r is necessarily odd. In that case, by similar arguments, we can prove that $\mathsf{vc}(C \setminus B) + |B| = (|C| + 1)/2 = (r + 1)/2 = \mathsf{vc}(C)$ and hence again B is a good set. $\square$

Lemma 3.11. *Let C be a cycle of length r and let $B \subset C$ of size $q \geq 2$ such that the clockwise distance between every consecutive pair of vertices in B is odd. Then B is a blocking set.*

Proof. Every component of $C \setminus B$ is a path with an even number p_i of vertices. Hence as in the proof the previous lemma, we have $\mathsf{vc}(C \setminus B) + |B| = \sum_{i=1}^{q} p_i/2 + q = \sum_{i=1}^{q} (p_i + 2)/2 = |C|/2 + q/2$.

If $q = 2$, then $|C|$ is even (as it is the sum of two odd numbers) and $\mathsf{vc}(C) = \frac{|C|}{2}$ and hence $\mathsf{vc}(C \setminus B) + |B| = \frac{|C|}{2} + 1 > \mathsf{vc}(C)$.

If $q > 2$, then $\frac{|C|}{2} + \frac{q}{2} \geq \frac{|C|+3}{2} \geq \frac{|C|+1}{2} + 1 \geq \mathsf{vc}(C) + 1$ and hence $\mathsf{vc}(C \setminus B) + |B| > \mathsf{vc}(C)$. Hence in all cases B is a blocking set. $\square$

Proof of Theorem 3.1: From Observation 2, no single vertex of any cycle forms a blocking set.

- For an odd cycle, let $B = \{a_{i_1}, \ldots, a_{i_p}\}$ be a minimal blocking set of size at least 4. Without loss of generality let us assume that $dist(a_{i_1}, a_{i_2})$ is odd. If $dist(a_{i_t}, a_{i_{t+1}})$ is even for all $2 \leq t \leq p-1$, then $dist(a_{i_p}, a_{i_1})$ is even as the length of cycle is odd. But then the distance between at most one consecutive pair of vertices is odd. Then B is a good set by Lemma 3.10. This is a contradiction. Then there exists $t \geq 2$ such that $dist(a_{i_t}, a_{i_{t+1}})$ is odd. Lets pick the smallest $t \geq 2$ such that $dist(a_{i_t}, a_{i_{t+1}})$ is odd. Then $dist(a_{i_2}, a_{i_{t+1}})$ is odd. Now consider $B' = \{a_{i_1}, a_{i_2}, a_{i_{t+1}}\}$. Clearly $B' \subset B$ and the clockwise distance between every consecutive pair in B' is odd. So, by Lemma 3.11, B' is a blocking set contradicting the minimality of B. Now clearly for a blocking set of size exactly three, by Lemma 3.10, clockwise distance between every consecutive pair in B' is odd.

- In an even cycle, suppose that $\{a_i, a_j\}$ is a blocking set. Then from Lemma 3.10, it is clear that $dist(a_i, a_j)$ and $dist(a_j, a_i)$ are odd. Suppose that there exists a minimal blocking set B of size at least three in an even cycle. Then from Lemma 3.10, we know that there are at least two vertices a_{i_j} and a_{i_t} in B such that $dist(a_{i_j}, a_{i_{j+1}})$ and $dist(a_{i_t}, a_{i_{t+1}})$ are odd (t could be $j + 1$). Then from Lemma 3.11, we have that the set $\{a_{i_j}, a_{i_{j+1}}\} \subset B$ is a blocking set contradicting the minimality of B.

This completes the proof. $\square$

Using Theorem 3.1, we see that the size of any "minimal blocking sets" in an odd cycle is exactly three. Based on this property, we partition the set of odd cycles (or the components in $G[F_2]$) into two parts, *bad component* and *nice component*. We move on to provide the formal definitions as follows.

Definition 3.3 (Bad Component and Nice Component)**.** *Let C be a cycle in $G[F_2]$. If there exists an independent set $A \subseteq S$ of size at most 3 such that $N_G(A) \cap C$ contains a blocking set in $G[C]$, then we call C a bad component. A component is said to be a nice component if it is not a bad component.*

Observation 3. *Let C be a connected component in $G[F_2]$. Then, $C \in \mathcal{B}_4$ if and only if C is a nice component in $G[F_2]$.*

Proof. ($\Rightarrow$) Suppose $C \in \mathcal{B}_4$ where C is a bad component. Then there exists $A \subseteq S$ such that $|A| > 3$ and $N_G(A) \cap C = C'$ which is a blocking set in C. Note that $|C'| \geq 3$ and $|A| \geq 4$. Then there exists $C'' \subseteq C'$ of size 3 (by Theorem 3.1) such that C'' is a minimal blocking set. As $|A| > 3$, there exists $A' \subseteq A$ such that $|A'| \leq 3$ and $C'' \subseteq N_G(A')$. Then $C \in \mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{B}_3$ which is a contradiction that $C \in \mathcal{B}_4$ and $C \notin \mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{B}_3$. So C is a nice component and the observation is true.
($\Leftarrow$) Let C be a nice component of $G[F_2]$ when C is not a bad component in $G[F_2]$. Then $C \notin \mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{B}_3$. So, $C \in \mathcal{B}_4$. ☐

Reduction Rule 3.8 (Nice Component Rule)**.** *Let C be a nice component in $G[F_2]$. Then we set $G' \leftarrow G \setminus C, S' \leftarrow S$ and $\ell' \leftarrow \ell - \mathsf{vc}(G[C])$.*

Lemma 3.12. *Reduction Rule 3.8 is safe and can be applied in polynomial time.*

Proof. First we prove that this reduction rule is safe. It suffices to prove that there exists a minimum vertex cover for G containing exactly $\mathsf{vc}(C)$ many vertices from C. Let D be a minimum vertex cover for $G \setminus C$. We claim that D can be completed to a minimum vertex cover of G by picking $\mathsf{vc}(C)$ many vertices from C.

Let $S' = S \setminus D$. We are done if we show that $N_G(S') \cap C = C'$ is a *good set*. In order to get a minimum vertex cover of G, we can first pick $N_G(S') \cap C$ and as it is a good set, we can complete it to get an optimum vertex cover of C by picking an overall $\mathsf{vc}(C)$ many vertices from C. Note that $G[S']$ is an independent set. Suppose for the sake of contradiction that after picking C', we see that if we want to complete it to a vertex cover of G, then we must pick at least $\mathsf{vc}(C) + 1$ vertices. Then C' is a blocking set. As C is an odd cycle, by Theorem 3.1, there exists a subset C'' of C' such that $|C''| = 3$ and C'' is a blocking set from Theorem 3.1. Then there exists an independent set $S'' \subseteq N_G(C'') \cap S'$ of size at most 3 such that $C'' \subseteq N_G(S'') \cap C$ which implies that C is a bad component, a contradiction to the fact that C is a nice component. So this reduction rule is safe.
Now we prove that this rule can be applied in polynomial time. By Definition 3.3, we know that a component $C \in G[F_2]$ is bad when there exists an independent set $A \subseteq S$ of size at most 3 such that $N_G(A) \cap C$ contains a blocking set in $G[C]$. So, C is nice component when for all $A \subseteq S$ of size at most 3, $N_G(A) \cap C$ is not a blocking set in $G[C]$. So for C being a bad component, we need to find an witness A in S of size at most 3 such that $N_G(A) \cap C$ forms a blocking set in $G[C]$. There are at most $(|S| + \binom{|S|}{2} + \binom{|S|}{3})$ choices of A and we can test it in polynomial time whether any such witness exists. So, this reduction rule can be implemented in polynomial time. ☐

Note that Reduction Rules 3.5 and 3.6 used $X \subseteq S$ such that X is an independent set of size at most 2 and X is not a Chunk. Now that we have bounded the size of a minimal blocking set of an odd cycle to 3, a similar rule on independent sets of size 3 would result in $|\mathcal{B}_3| = \mathcal{O}(k^4)$ and eventually a kernel upper bound with $\mathcal{O}(k^6)$ vertices. But we produce a smaller kernel using *Expansion Lemma* by constructing an auxiliary graph. This approach improves the kernel upper bound by actually deleting some 'redundant' components in $\mathcal{B}_3$.

The bipartite graph $H = (S_3, \mathcal{B}_3, E)$, we construct has one part of the vertex set as S_3, which is the set of all independent sets of size 3 from S. The edge set $E(H)$ is defined as follows. $E(H) = \{(I, L) | I \in S_3, L \in \mathcal{B}_3$, there exists $B \subseteq V(L)$ such that B is a blocking set of size 3 in L and $B \subseteq N_G(I)\}$.

Lemma 3.13. *Auxiliary graph H can be constructed in polynomial time.*

Proof. To identify cycles in $G[F_2]$ that are in $\mathcal{B}_3$, we need to to identify and rule out the cycles in $\mathcal{B}_1$ and $\mathcal{B}_2$. To check if a cycle C is in $\mathcal{B}_1$, we have to check if there exists a vertex $x \in S$ such that $N_G(x) \cap V(C)$ is a blocking set in C. In this way, we first identify the set of cycles that are in $\mathcal{B}_1$.

This completes the construction of $\mathcal{B}_1$. Next, among the remaining cycles, we check if there exists a nonadjacent pair $(x,y) \in \binom{S}{2}$ such that $N_G(x) \cup N_G(y)$ contains a blocking set in some cycle C in $G[F_2]$. In this way, we construct the set $\mathcal{B}_2$. Now, when Reduction Rule 3.8 is not applicable, then we know that the remaining cycles must be in $\mathcal{B}_3$. We can check using Theorem 3.1 whether a set of vertices in a cycle is a blocking set or not. After that we construct the graph H as described. So, the auxiliary graph H can be constructed in polynomial time. $\square$

Reduction Rule 3.9 (Expansion Lemma Rule)**.** *If $|\mathcal{B}_3| \geq 4|S_3|$, then apply Expansion Lemma 2.1 with $q = 4$ from S_3 to $\mathcal{B}_3$ to get $A \subseteq S_3, B \subseteq \mathcal{B}_3$ with $A, B \neq \emptyset$ such that $N_H(B) \subseteq A$ and there is a 4-expansion from A to B. Associated with every $(x,y,z) \in A$, there are 4 distinct cycles in B. Pick one of those 4 cycles for each such $\{x,y,z\} \in A$, delete them after decreasing ℓ by the size of the sum of their optimum vertex cover sizes. I.e. Let $C_{p_1}, \ldots, C_{p_{|A|}}$ be collection of such cycles. Then set $G' \leftarrow G \setminus (C_{p_1} \cup \ldots \cup C_{p_{|A|}}), S' \leftarrow S$ and $\ell' \leftarrow \ell - (\sum_{i=1}^{|A|} \mathsf{vc}(C_{p_i}))$.*

Lemma 3.14. *Reduction Rule 3.9 is safe and can be applied in polynomial time.*

Proof. It follows from Lemma 3.13 and Lemma 2.1 that the reduction rule can be implemented in polynomial time. Now we prove why the reduction rule is safe. We prove this in two steps. First we show that there exists a minimum vertex cover of G' containing at least one vertex from every independent set in A. Then we will show that this implies that there exists a vertex cover of G that has exactly $\mathsf{vc}(C)$ many vertices from every $C \in B$ we have deleted.

The intuition for the first step is that due to the expansion property, every triple in A contains a private set of 4 cycles in which it has a blocking set as a neighbor. Out of these 4 cycles, we have deleted one and so at least 3 remain. If no vertex of the triple is in the solution, then their neighbors (that form a blocking set in each of the 3 cycles) need to be in the solution resulting in 3 extra vertices in the vertex cover than what are needed to cover the edges of the three cycles (by the definition of blocking set). Instead, we are better of picking all vertices of the triple and choosing only the optimum from each of the 3 cycles.

Once we establish this, the second step follows as now for every cycle deleted, if we delete the vertices (from A) that are already in the vertex cover, there is no blocking set as the triples are not in $\mathcal{B}_1$ or $\mathcal{B}_2$. Hence we can pick enough to cover the edges of those cycles alone to complete the vertex cover.

We proceed with the formal proof that there exists a minimum vertex cover of G' containing at least one vertex from every independent set in A. Suppose not. Then let D be a minimum vertex cover of G' and suppose $A' \subseteq A$ is a set of independent triples that have no intersection with D (i.e. $D \cap V(A') = \emptyset$ where $V(A') = \bigcup_{\{x,y,z\} \in A'} \{x,y,z\}$). By the 4-expansion property, for each such triple $T = \{x,y,z\} \in A'$, there are at least 4 distinct odd cycles and as we have removed one of them from G, we will still have at least 3 cycles, say $C_{T_1}, C_{T_2}, C_{T_3}$ such that each of them contains a blocking set which is in D (as no vertex of A' is in D). Thus for every $T \in A'$ and for every $i \in \{1,2,3\}$, $|D \cap C_{T_i}| \geq \mathsf{vc}(C_{T_i}) + 1$. Let $C_{A'}$ be the set of all distinct $3|A'|$ such cycles. We construct D' from D as follows. $D' = (D \setminus V(C_{A'})) \cup V(A')$. Let $S' = S \setminus D'$. Note that $S' \cap V(A') = \emptyset$.

We claim that that for all $C \in C_{A'}, N_G(S') \cap C$ is a good set in C. First we show that we are done if we prove the claim. This is because, then we can add $\sum_{C \in C_{A'}} \mathsf{vc}(C_{A'})$ many vertices to get a vertex cover to get an overall vertex cover D_1 for G'. Then,

$$|D_1| \leq |D| - (\sum_{C \in C_{A'}} (\mathsf{vc}(C_{A'}) + 1) + 3|A'| + \sum_{C \in C_{A'}} \mathsf{vc}(C_{A'})$$

$$= |D| - |C_{A'}| + 3|A'| \leq |D| - 3|A'| + 3|A'| = |D|$$

Hence, we get a minimum vertex cover D_1 which contains at least one vertex from every $T \in A$. Now, we move on to prove the claim that for every $C \in C_{A'}, N_G(S') \cap C$ is a good set. Suppose not. Let C be that cycle in $C_{A'}$ such that $N_G(S') \cap C$ contains a minimal blocking set $L \subseteq C$ of size 3. In fact, there is also an independent set $S'' \subseteq S', |S''| \leq 3$ such that $N_G(S'') \supseteq L$. As $C \in \mathcal{B}_3$ and $\mathcal{B}_3$ is

disjoint from each of $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_4$, we get that $|S''| = 3$. Then as $N_H(B) \subseteq A$, $S'' \in A$. But D' contains a vertex from every triple of A which contradicts the fact $S'' \cap D' = \emptyset$.
Thus we have shown that there exists a minimum vertex cover of G' that contains at least one vertex from every triple of A. Let D_3 be such a vertex cover, and let $S'_3 = S \setminus D_3$.

We claim that $N_G(S'_3) \cap C$ is a good set for any C that was deleted. We are done if we prove the claim as then we can add $\mathsf{vc}(C)$ many vertices from every such C to get a minimum vertex cover for G. Now, suppose for some C (that was deleted) $N_G(S'_3) \cap C$ is a blocking set. Note that $C \in B$ and $B \subseteq \mathcal{B}_3$. So, $C \in \mathcal{B}_3$. Then, by the Expansion Lemma (Lemma 2.1), we know that $N_H(C) \subseteq A$. Moreover any vertex of A is an independent set of size 3 in G. As $C \in \mathcal{B}_3$, there does not exist $x, y \in S, (x, y) \notin E(G)$ and $C \cap (N_G(x) \cup N_G(y))$ contains a blocking set in C. Therefore, there exists $\{a, b, c\} \subset S'_3$ such that $N_G(\{a, b, c\}) \cap C$ is a blocking set. It means that $((a, b, c), C) \in E(H)$. But, then $(a, b, c) \in A$. But by our choice of D_3, at least one of a, b, c is in D_3. This is a contradiction. Therefore, for any C which is deleted, $N_G(S'_3) \cap C$ is a good set. This completes the proof that the reduction rule is safe. □

Corollary 3.5. *If Reduction Rule 3.9 is not applicable, then $|\mathcal{B}_3| \leq 4\binom{k}{3}$.*

Finally the following lemma follows from Reduction Rule 3.8 (as if this rule is not applicable, every component is bad and $\mathcal{B}_4 = \emptyset$) and Corollaries 3.3, 3.4 and 3.5.

Lemma 3.15. *If none of the Reduction Rules 3.1 to 3.9 are applicable, then the number of components in $G[F_2]$ is $\mathcal{O}(k^3)$.*

3.4 Bounding $G[F_2]$ and Putting things together

We know from Lemma 3.15 that the number of odd cycles in $G[F]$ is $\mathcal{O}(k^3)$. We also have from Lemma 3.8 that the number of vertices in paths and even cycles in $G[F]$ is $\mathcal{O}(k^3)$. So, the number of connected components in $G[F]$ that have cycles is $\mathcal{O}(k^3)$. So, $G[F]$ has a feedback vertex set of size $\mathcal{O}(k^3)$. As $|S| \leq k$ and $G[F]$ has a feedback vertex set with $\mathcal{O}(k^3)$ vertices, we have that G has a feedback vertex set with $\mathcal{O}(k^3)$ vertices. Then, we can apply Theorem 2.1 on this graph and can get a kernel with $\mathcal{O}(k^9)$ vertices. But in order to improve the upper bound further, we utilize the graph structure more carefully and provide an improved upper bound on the number of vertices in $G[F_2]$.

Finally, we bound the total number of edges in $G[F_2]$, i.e. in the odd cycles of $G[F]$. Let c be the number of odd cycles that remain in $G[F_2]$ after all the earlier reduction rules have been applied. From Lemma 3.15, c is $\mathcal{O}(k^3)$.

Lemma 3.16. *Let G be a graph in which every component is an odd cycle and there are c components in G and M be a maximum matching in G. Given $X \subseteq V(G)$, let X contains both the endpoints of $r + c$ edges of M. Then $|X| + \mathsf{vc}(G \setminus X) \geq \mathsf{vc}(G) + r$.*

Proof. As every component of G is an odd cycle, so there exists a vertex from each of the components that is not matched by M. Let A be the set of vertices that are not matched by M. Let $B = V(G) \setminus A$. Also all vertices of B are matched by M while $G[B]$ is bipartite. So we have that $\mathsf{vc}(G) = |M| + c = \mathsf{vc}(G[B]) + c$. Let $X \subseteq V(G)$ such that X contains both the endpoints of $r + c$ edges of M. Then we have that

$$|X| + \mathsf{vc}(G \setminus X) = |X \cap A| + |X \cap B| + \mathsf{vc}(G \setminus X)$$

$X \cap B$ contains both endpoints of $r + c$ edges of M. And $G[B]$ is a bipartite graph and M saturates all the vertices of B. Also $\mathsf{vc}(G \setminus X) \geq \mathsf{vc}(G[B \setminus X])$ as $G[B \setminus X]$ is an induced subgraph of $G \setminus X$. So we have that $|X \cap B| + \mathsf{vc}(G \setminus X) \geq |X \cap B| + \mathsf{vc}(G[B \setminus X]) \geq \mathsf{vc}(G[B]) + r + c$ by Lemma 3.6. As $\mathsf{vc}(G[B]) + c = \mathsf{vc}(G)$, we have that

$$|X \cap A| + |X \cap B| + \mathsf{vc}(G \setminus X) \geq \mathsf{vc}(G[B]) + r + c = \mathsf{vc}(G) + r$$

This completes the proof of the lemma. □

Now, when the Reduction Rules 3.5, 3.6 are not applicable, we have the following statements using Lemma 3.16.

Lemma 3.17. *Let M_1 be a* maximum matching *in $G[F_2]$ and c be the number of components in $G[F_2]$. If Reduction Rules 3.5 and 3.6 are not applicable, then the following statements are true.*

1. *For every $x \in S$, $N_G(x)$ contains both the endpoints of at most $|S| + c$ edges in M_1.*
2. *For every $x, y \in S, (x, y) \notin E(G)$, $N_G(\{x, y\})$ contains both the endpoints of at most $|S| + c$ edges in M_1.*

Proof. We prove the statements in the given order.

1. Suppose that $N_G(x)$ contains both the endpoints of $r + c$ edges in M_1 where $r \geq |S| + 1$. Let $N_G(x) \cap F = A$. By Lemma 3.16, we know that $|N_G(x) \cap F_2| + \mathsf{vc}(G[F_2 \setminus N_G(x)]) = |A \cap F_2| + \mathsf{vc}(G[F_2 \setminus A]) \geq \mathsf{vc}(G[F_2]) + r$. For our case, $r \geq |S| + 1$. So, $\mathsf{vc}(G[F_2 \setminus N_G(x)]) + |N_G(x) \cap F_2| \geq \mathsf{vc}(G[F_2]) + |S| + 1$. Now, we have that
$$|N_G(x) \cap (F_0 \cup F_1)| + \mathsf{vc}(G[(F_0 \cup F_1) \setminus N_G(x)]) \geq \mathsf{vc}(G[F_0 \cup F_1])$$
We know that
$$|N_G(x) \cap F| + \mathsf{vc}(G[F \setminus N_G(x)])$$
$$= |N_G(x) \cap F_2| + |N_G(x) \cap (F_0 \cup F_1)| + \mathsf{vc}(G[F_2 \setminus N_G(x)]) + \mathsf{vc}(G[(F_1 \cup F_0) \setminus N_G(x)])$$
So, $|N_G(x) \cap F| + \mathsf{vc}(G[F \setminus N_G(x)]) \geq \mathsf{vc}(G[F_2]) + |S| + 1 + \mathsf{vc}(G[F_0 \cup F_1]) = \mathsf{vc}(G[F]) + |S| + 1$. Then $\{x\}$ is not a Chunk. So, Reduction Rule 3.5 becomes applicable which is a contradiction.
2. Suppose that for some nonadjacent pair $(x, y) \in \binom{S}{2}$, we have that $N_G(\{x, y\})$ contains both the endpoints of $r + c$ edges in M_1. Let $N_G(\{x, y\}) \cap F = A$. For our case $r \geq |S| + 1$. We have that, $\mathsf{vc}(G[F \setminus N_G(\{x, y\})]) + |N_G(\{x, y\}) \cap F| = |A \cap F| + \mathsf{vc}(G[F \setminus A])$. Now, we know that $|A \cap F| + \mathsf{vc}(G[F \setminus A]) = |A \cap F_2| + |A \cap (F_0 \cup F_1)| + \mathsf{vc}(G[F_2 \setminus A]) + \mathsf{vc}(G[(F_0 \cup F_1) \setminus A])$. By Lemma 3.16, $|A \cap F_2| + \mathsf{vc}(G[F_2 \setminus A]) \geq \mathsf{vc}(G[F_2]) + r \geq \mathsf{vc}(G[F_2]) + |S| + 1$. We also know that $|A \cap (F_0 \cup F_1)| + \mathsf{vc}(G[(F_0 \cup F_1) \setminus A]) \geq \mathsf{vc}(G[F_0 \cup F_1])$. So, we have that
$$|A \cap F| + \mathsf{vc}(G[F \setminus A]) = |A \cap F_2| + |A \cap (F_0 \cup F_1)| + \mathsf{vc}(G[F_2 \setminus A]) + \mathsf{vc}(G[(F_0 \cup F_1) \setminus A])$$
$$\geq \mathsf{vc}(G[F_2]) + \mathsf{vc}(G[F_0 \cup F_1]) + |S| + 1 = \mathsf{vc}(G[F]) + |S| + 1$$
Then, $\{x, y\}$ is not a Chunk. So, Reduction Rule 3.6 becomes applicable which is a contradiction.

This completes the proof of the lemma. ☐

Lemma 3.18. *When Reduction Rules 3.1 to 3.9 are not applicable, the number of vertices in $G[F_2]$ is $\mathcal{O}(k^5)$.*

Proof. By Reduction Rule 3.7, for every edge $(u, v) \in E(G[F])$ (also for every $(u, v) \in M_1$), there is either a vertex $x \in S$ such that $x \in N(u) \cap N(v)$ or there exists a pair of nonadjacent vertices $x, y \in S$ such that $x \in N(u), y \in N(v)$ and $N_G(u) \cap N_G(v) \cap S = \emptyset$. In the first case, we associate the vertex x with the edge (u, v). In the second case, we associate the pair of nonadjacent vertices (x, y). Let M_1 be a maximum matching in $G[F_2]$ and c be the number of components in $G[F_2]$. By Lemma 3.17, for every $x \in S$, x is adjacent to both endpoints of at most $k + c$ edges of M_1. And every pair of nonadjacent vertices $x, y \in S$, x and y together are adjacent to both endpoints of at most $k + c$ edges of M_1. Therefore, $|M_1| \leq (k + \binom{k}{2})(k + c) = \mathcal{O}(k^5)$ as c is $\mathcal{O}(k^3)$ by Lemma 3.15. The number of vertices not matched by M_1 is at most the number of components in $G[F_2]$. So it follows that $|V(G[F_2])| = 2|M_1| + c$ which is $\mathcal{O}(k^5)$. ☐

We know from Remark 1 that Reduction Rules 3.1 to 3.6 do not increase the parameter. Now, note that Reduction Rules 3.7, 3.8 and 3.9 do not add any vertex in S as they perform only local operation in F. So, the size of the modulator (or the parameter) cannot increase by any of the reduction rule. So the following theorem is an immediate consequence of Lemma 3.8 and Lemma 3.18. The bound on the number of edges follows because the number of edges in $G[S]$ is $\mathcal{O}(k^2)$, the number of edges in $G[F]$ is $\mathcal{O}(k^5)$ and the number of edges between F and S can be at most $\mathcal{O}(k^6)$.

Theorem 3.2. VC-2-Mod *has a kernel with $\mathcal{O}(k^5)$ vertices and $\mathcal{O}(k^6)$ edges.*

4 Vertex Cover Parameterized by Modulator to Trees and Cycles

In this section, we explain how our ideas from the previous section can be used to extend to a polynomial kernel when Vertex Cover problem is parameterized by the size of a set whose deletion results in a disjoint union of trees and cycles. Before, that we recall the definition of the term *pseudo-forest.*

Definition 4.1 (Pseudo-Forest)**.** *An undirected graph G is said to be a* pseudo-forest *if every connected component of G has at most one cycle.*

Vertex Cover parameterized by the size of a modulator to a pseudo-forest has a kernel with $\mathcal{O}(k^{12})$ vertices [16, 17]. Now we consider a parameterization where the modulator $S \subseteq V(G)$ is a deletion set to a graph where every component is either a tree or an induced cycle and show that Vertex Cover with such a parameterization has a kernel with $\mathcal{O}(k^9)$ vertices. Note that $G \setminus S$ in this case is a special case of a pseudo-forest. Formal definition of our problem is as follows.

Vertex Cover parameterized by Modulator to Trees and Cycles (VC-Trees-Cycles)
Input: An undirected graph G, $S \subseteq V(G)$ of size at most k such that every component of $G[V(G) \setminus S]$ is either a cycle or a tree and an integer ℓ.
Parameter: k
Question: Does G have a vertex cover of size at most ℓ?

Let $F = V(G) \setminus S$. We partition F as $T \uplus J$ where $G[T]$ is the graph whose every component is a *tree* and $G[J]$ is the graph where every component is an induced cycle.

We first apply Reduction Rules 3.1 to 3.6 (Isolated Vertex Rule, Self Loop Rule, Degree 1 Rule, Degree 2 Rule, Non-Chunk-Rule-I and Non-Chunk-Rule-II) to the instance (G, S, ℓ). When they are not applicable, we apply Reduction Rule 3.7 (Edge Rule) on the edges of the components that are cycles. After the exhaustive applications of these rules, we apply Reduction Rule 3.8 (Nice Component Rule) and 3.9 (Expansion Lemma Rule) exhaustively on the odd cycles of $G[S \cup J]$. Let (G', S', ℓ') be the resulting equivalent parameterized problem. We first argue about the safety of the reduction rules despite the new modulator.

Lemma 4.1. *Reduction Rules 3.1 to 3.6 are safe. Moreover, the existence of T does not affect the safety of the Reduction Rules 3.7, 3.8 and 3.9 when applied on $G[S \cup J]$.*

Proof. We basically argue that the existence of $G[T]$ does not affect the safety of the applied Reduction Rules. From Remark 1, we know that Reduction Rules 3.1 to 3.6 are applicable for any instance (G, S, ℓ) where vertex cover is polynomial time solvable on $G \setminus S$ and $G \setminus S$ is hereditary as well as closed under contraction of edges. So, Reduction Rules 3.1 to 3.6 are safe even with $G[T]$. Also from Remark 1, we know that those reduction rules do not increase the size of the modulator. Now as we apply Reduction Rule 3.7 (Edge Rule) on the edges of cycles, the premise of applying the reduction rule is satisfied. Also for Nice Component Rule (Reduction Rule 3.8) and Expansion Lemma Rule (Reduction Rule 3.9), the premise is satisfied when they are applied to odd cycles in $G[J]$. There is no edge between a vertex of odd cycle and a tree. And as those rules are applied locally (and they have no interaction with the components of T), existence of T does not affect the safety of Reduction Rule 3.7. Reduction Rule 3.7, 3.8 do not interact with the vertices of T. So safety of them are not affected. Now when we apply Reduction Rule 3.9, then we do not delete any vertex from S. We only delete some components from J. As such components do not interact with vertices of T, safety of Reduction Rule 3.9 is not affected. □

Note that the adjacency between the vertices of S' and T will remain the same as they were in S as $S' \subseteq S$ as we only delete vertices from S. So for any $x \in S', u \in T$, as x is also in S, we have that $(x, u) \in E(G)$ if and only if $(x, u) \in E(G')$. If some vertex from S gets deleted during the construction of S', then that vertex gets deleted from the whole graph.

Notice that Reduction Rule 3.7 was also applicable to the edges of those components of $G \setminus S$ that are paths. But, for this problem, we apply them only for the cycles. Similar to the proof of Lemma 3.8, we can prove the following claim.

Lemma 4.2. *When Reduction Rules 3.1 to 3.9 are not applicable, the number of vertices in the even cycles of $G[J]$ is $\mathcal{O}(k^3)$ and the number of odd cycles in $G[J]$ is $\mathcal{O}(k^3)$.*

Every even cycle has at least 4 vertices. So, the number of even cycles in $G'[J]$ is $\mathcal{O}(k^3)$ by Lemma 4.2. We also get from Lemma 4.2 that the number of odd cycles in $G'[J]$ is also $\mathcal{O}(k^3)$.

Now we transform the instance (G', S', ℓ') to instance (G', S'', ℓ') by modifying S' to get S'' as follows. Pick any arbitrary vertex from every component in $G' \setminus S'$ that has a cycle and let J' be the set of chosen vertices. Let $F' = V(G') \setminus S'$. Now we set

- $S'' \leftarrow S' \cup J'$.
- $F'' \leftarrow F' \setminus J'$.

By Lemma 3.15, we know that $|J'| = \mathcal{O}(k^3)$. Now, we see that $G'[F'']$ is a forest. It means that S'' is a Feedback Vertex Set with $|S''|$ which is $\mathcal{O}(k^3)$. The resulting problem now is the Vertex Cover parameterized by the size of a feedback vertex set for which there is a kernel with $\mathcal{O}(|S''|^3)$ vertices from Theorem 2.1 which results in an $O(k^9)$ vertex kernel for our problem. Thus we have the following theorem.

Theorem 4.1. VC-Trees-Cycles *has a kernel with $\mathcal{O}(k^9)$ vertices.*

5 Vertex Cover parameterized by bounded cluster vertex deletion set

Now, we consider the Vertex Cover problem when parameterized by the size of degree-1-modulator (or equivalently the size of a subset whose deletion results in a graph of degree at most one). Here the resulting graph is a collection of isolated vertices and edges, i.e. disjoint union of cliques of size at most two. In fact, what we will give is a kernel for VC-param-d-CVD.

Vertex Cover parameterized by d-CVD (VC-param-d-CVD) **Input:** An undirected graph G, $S \subseteq V(G)$ of size at most k such that every connected component of $G[V(G) \setminus S]$ is a clique with at most d vertices and an integer ℓ. **Parameter:** k **Question:** Does G have a vertex cover of size at most ℓ?

Recall that we use S to denote the set of vertices to be deleted such that each component of $G \setminus S$ is a clique and each such clique has at most d vertices. We use $F = V(G) \setminus S$. If there is no bound on the size of the components in F, then it is known that the problem has no polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. In fact, Bodlaender et al. [2] showed this infeasibility even when F is a clique (with no restriction on the number of vertices).

For this problem, we would also need the notion of a *hypergraph*. A *hypergraph*, H, consists of vertex set $V(H)$ and hyperedge set $E(H)$, where every *hyperedge* is a subset of $V(H)$. Given a hyperedge $e \in E(H)$, by $V(e)$, we denote the subset of $V(H)$ that e comprises of. Given $X \subseteq V(H)$, an induced hypergraph on X is denoted by $H[X]$ and it consists of all the edges $h \in E(H)$ such that $h \subseteq X$. Similarly for $X \subseteq V(H)$, we use $H \setminus X$ to denote hypergraph $H[V(H) \setminus X]$.

Definition 5.1 (Independent Set in a hypergraph)**.** *A set of vertices $A \subseteq V(H)$ is said to be an independent set in a hypergraph if no hyperedge is contained in $H[A]$.*

Definition 5.2 (Vertex Cover in a hypergraph)**.** *A set of vertices $A \subseteq V(H)$ is said to be a vertex cover in hypergraph if for every hyperedge $e \in E(H)$, $A \cap V(e) \neq \emptyset$. A vertex cover in a hypergraph is also known as a hitting set.*

Now for this problem we partition the components of $G[F]$ into two types. As every component of $G[F]$ is a clique, we call them a *good clique* and a *bad clique*.

Definition 5.3 (Bad Clique and Good Clique). *A component C in $G[F]$ is said to be a* bad clique *if there exists an independent set A of at most d vertices from S such that $V(C) \subseteq N_G(A)$. A component C is said to be a* good clique *if it is not a bad clique.*

Now, we proceed to state the list of reduction rules for this problem. As a preprocessing, we only require that the isolated vertices are removed. So, we do not need to even make the graph minimum degree three. In case a component in $G[F]$ is an isolated vertex in G, then it is a good clique. The next reduction rules have similarity with Reduction Rules 3.5, 3.6 and 3.8 of Section 3. But as it is applied for different types of graphs, we state them here for completion.

Reduction Rule 5.1. *Let C be a component of $G[F]$ where C is a good clique, then*

- $G' \leftarrow G \setminus C$.
- $S' \leftarrow S$.
- $\ell' \leftarrow \ell - |V(C)| + 1$.

Lemma 5.1. *Reduction Rule 5.1 is safe.*

Proof. ($\Rightarrow$) Let D be a vertex cover of G of size at most ℓ. So, $D \setminus V(C)$ is a vertex cover of G'. Moreover, D must contain at least $|V(C)| - 1$ vertices as C is a clique in $G[F]$ (hence in G). So, $|D \setminus V(C)| \leq |D| - (|V(C)| - 1) = |D| - |V(C)| + 1$. Hence, $D \setminus V(C)$ is a vertex cover of G' of size at most ℓ'.

($\Leftarrow$) Let D' be a vertex cover of G' of size at most ℓ'. Let $S' = S \setminus D$. S' is an independent set. If we can prove that $V(C) \not\subseteq N_G(S')$, then we are done. Because, in that case, at least one vertex in C is not a neighbor to S'. So, we can add $|V(C)| - 1$ vertices into D' and get a vertex cover of size at most ℓ for G. So, we will prove that $V(C) \not\subseteq N_G(S')$. Suppose $V(C) \subseteq N_G(S')$. If S' has at most d vertices, then C is a bad clique which is a contradiction. Otherwise $|S'| > d$ and $V(C) \subseteq N_G(S')$. So, there exists $A' \subseteq S'$ consisting of at most d vertices such that and $V(C) \subseteq N_G(A')$. Then also C is a bad clique which is a contradiction. So, $V(C) \not\subseteq N_G(S')$ and the reduction rule is safe. □

Observe that every component C in $G[F]$, that contains a vertex which has no neighbor in S, is a good clique. When Reduction Rule 5.1 is not applicable, every vertex in F has at least one neighbor in S. Recall that in Section 3, we defined Chunk. We will use the property of Chunk to design the next reduction rule. The next reduction rule has similarity with Reduction Rule 3.5 and 3.6 in Section 3.

Reduction Rule 5.2. *If there exists an independent set A of at most $d - 1$ vertices from S such that A is not a* Chunk*, then do the following:*

1. *If $|A| = 1$, then we set $G' \leftarrow G \setminus A, S \leftarrow S \setminus A$ and $\ell' \leftarrow \ell - 1$.*
2. *If $2 \leq |A| \leq (d - 1)$, then add hyperedge $\{x_1, \ldots, x_{|A|}\}$ to get the graph G' where $A = \{x_1, \ldots, x_{|A|}\}$. We set $S' \leftarrow S$ and $\ell' \leftarrow \ell$.*

Lemma 5.2. *Reduction Rule 5.2 is safe.*

Proof. We prove the correctness of this reduction rule in the given order. Let D be a minimum vertex cover of G and $A \subseteq S$ is an independent set of size at most d such that A is not a Chunk. By Corollary 3.2, we know that $D \cap A \neq \emptyset$. Then we have the following cases.

1. If $|A| = 1$, we have that $A \subseteq D$. So, any minimum vertex cover of G must contain A and the reduction rule is safe.
2. If $2 \leq |A| \leq d - 1$, we have that $D \cap A \neq \emptyset$. In other words, any minimum vertex cover of G must contain at least one vertex from A. Adding an hyperedge that contains exactly the set of vertices in A captures this constraint. So, the reduction rule is safe.

This completes the proof of the lemma. $\square$

Note that as we have added hyperedges in S of size at least three, we need the following domination rules. Note that hyperedges consisting of more than two vertices are present only in S. So, for any vertex $x \in S$, we can use $N_G(x) \cap F = \{y \in F | (x,y) \in E(G)\}$ and for any vertex $u \in F$, we can use $N_G(u) \cap S = \{v \in S | (u,v) \in E(G)\}$. For every vertex x, denote $HE(x) = \{e \in E(G) | x \in V(e)\}$. For every hyperedge e, the set of vertices that are present in the hyperedge e is denoted by $V(e)$. We have the following *domination rule* that are due to Abu-Khazm [1]. Note that when we delete a vertex u from a hypergraph, we delete all the hyperedges that are in $HE(u)$ as well unless specified explicitly.

Reduction Rule 5.3. *If there are two hyperedges e_1, e_2 such that $V(e_1) \subseteq V(e_2)$, then delete e_2 from G to get the graph G'. We set $S' \leftarrow S$ and $\ell' \leftarrow \ell$.*

We will need the following property of Chunk for this problem and this property is specially used to prove the number of components in $G[F]$.

Proposition 1. *Let A be any independent set consisting of at most d vertices from S. If $N_G(A) \cap F$ contains all vertices of at least $|S|+1$ components in $G[F]$, then A is not a* Chunk.

Proof. By Defintion 3.1, an independent set X of size at most d from S is called a Chunk when $|N_G(X) \cap F| + \mathsf{vc}(G[F \setminus N_G(X)]) \leq \mathsf{vc}(G[F]) + |S|$. Let t be the number of components in $G[F]$. Then $\mathsf{vc}(G[F]) = |F| - t$. Now, $|N_G(A) \cap F| + \mathsf{vc}(G[F \setminus N_G(A)] = |F| - t + r = \mathsf{vc}(G[F]) + r$ if $N_G(A) \cap F$ contains all vertices of r components in $G[F]$. By the precondition of the proposition, $r \geq |S|+1$. So, $|N_G(A) \cap F| + \mathsf{vc}(G[F \setminus N_G(A)] = |F| - t + r \geq |F| - t + |S| + 1 = \mathsf{vc}(G[F]) + |S| + 1$. By our assumption, A is an independent set and has at most d vertices. So A is not a Chunk. $\square$

We partition the set of bad cliques of $G[F]$ (that are also components of $G[F]$) into two parts Z_1 and Z_2 as follows.

- $Z_1 = \{C | C$ is a bad clique in $G[F]$ such that there exists an independent set $A \subseteq S$ of at most $d-1$ vertices such that $V(C) \subseteq N_G(A) \cap F\}$.
- $Z_2 = \{C | C$ is a bad clique in $G[F]$ for which there exists an independent set $A \subseteq S$ of at most d vertices such that $V(C) \subseteq N_G(A) \cap F\} \setminus Z_1$.

We have the following lemma when the above reduction rules are not applicable.

Lemma 5.3. *When Reduction Rules 5.2 and 5.3 are not applicable on an instance (G, S, ℓ), then $|Z_1| \leq k \sum\limits_{i=1}^{d-1} \binom{k}{i}$ where $|S| \leq k$.*

Proof. By the definition of Z_1, for every component $C \in Z_1$, there exists an independent set A of size at most $d-1$ of S such that $V(C) \subseteq N_G(A)$. As Reduction Rule 5.2 is not applicable, every independent set A of size at most $d-1$ is a Chunk. For any independent set A of size at most $d-1$ in S, $N_G(A)$ contains all vertices of at most k components by Proposition 1. So the number of components in Z_1 is at most $|S| \sum\limits_{i=1}^{d-1} \binom{|S|}{i} \leq k \sum\limits_{i=1}^{d-1} \binom{k}{i}$. $\square$

Let C be a component in Z_2 consisting of d vertices. Then there exists an independent set $A \subseteq S$ of exactly d vertices such that $V(C) \subseteq N_G(A)$. Now, we construct a bipartite graph $H(S_B, Z_2, J)$ where $S_B = \{X \subseteq S | X$ is an independent set in G and $|X| = d\}$. For every $A \in S_B$, for every $C \in Z_2$, we add an edge between A and C only when $V(C) \subseteq N_G(A) \cap F$.

Reduction Rule 5.4. *If $|Z_2| \geq (d+1)|S_B|$, then apply q-Expansion Lemma (Lemma 2.1) with $q = d+1$ from S_B to Z_2 to obtain $P_B \subseteq S_B, Q_B \subseteq Z_2$ such that $N_H(Q_B) \subseteq P_B$. For every $X \in P_B$, add the hyperedges of the form $\{x_1, \ldots, x_d\}$ where $X = \{x_1, \ldots, x_d\}$. Let the new graph be G'. We set $S' \leftarrow S$ and $\ell' \leftarrow \ell$ (see Figure 3 for an illustration).*

Lemma 5.4. *Reduction Rule 5.4 is safe.*

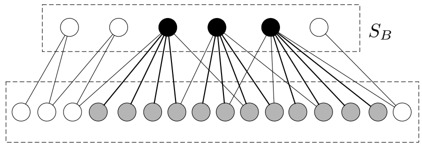

Figure 3: An illustration of Reduction Rule 5.4 when $q = 4$.

Proof. Note that P_B is a collection of d-tuples from S. All those tuples are independent sets. It suffices to prove that any minimum vertex cover of G contains at least one vertex from every d-tuple from P_B. Suppose not. Let D be a minimum vertex cover of G for which $X_1, \ldots, X_b \in P_B$ $(b > 0)$ such that $D \cap X_i = \emptyset$. For every $X_j \in P_B$ (where $j \in [b]$), there are $C_{j,1}, \ldots, C_{j,d+1} \in N_H(X_j) \cap Q_B$. Moreover, for every $i \in [d+1]$, $V(C_{j,i}) \subseteq D$. We construct D' from D as follows. We add $D' = D \cup (\bigcup_{i \in [b]} V(X_i))$ into D. So $|D'| = |D| + bd$. Now, for every $A \in P_B, A \cap D' \neq \emptyset$. Now, let $S' = S \setminus D'$. We claim that for every component $C \in Q_B$, $|N_G(S') \cap V(C)| < |V(C)|$. Suppose not. Then, let C' be a component in Q_B such that $V(C) \subseteq N_G(S')$. If $|S'| < d$, then $C' \in Z_1$. This is a contradiction as $C' \in Z_2$ by assumption where $Z_2 \cap Z_1 = \emptyset$. If $|S'| \geq d$, then there exists $A' \subseteq S'$ for which $|A'| = d$ and $V(C') \subseteq N_G(A')$. Then, $A' \in N_H(C')$. As $C' \in Q_B$, $A' \in P_B$ which is a contradiction to the fact that $D' \cap A' \neq \emptyset$. So, for every component $C \in Q_B$, $|N_G(S') \cap V(C)| < |V(C)|$. Now, for every component $C \in Q_B$, if $V(C) \subseteq D'$, then remove one vertex $p \in V(C)$ from D' such that $p \notin N_G(S')$. In this process, we construct D''. Note that we have removed $b(d+1)$ vertices from D' to get D''. So, $|D''| = |D'| - b(d+1) = |D| + bd - bd - b = |D| - b$. As $b > 0$, $|D'| < |D|$. This is a contradiction to the fact that D is a minimum vertex cover. So, any minimum vertex cover intersects every d-tuple in P_B. Hence the reduction rule is safe. □

The following Lemma is to justify that the above mentioned reduction rules can be implemented in polynomial time.

Lemma 5.5. *Reduction Rules 5.1, 5.2, 5.3, 5.4 can be implemented in polynomial time.*

Proof. Reduction Rule 5.1 requires us to check whether some component C a is good clique or not. By Definition 5.3, a component C is a bad clique when there exists an independent set A of size at most d in S such that $V(C) \subseteq N_G(A)$. As d is a constant it is possible to determine in $\mathcal{O}(k^d n^{\mathcal{O}(1)})$ time (by brute forcing over all possible subsets of S that are of size at most d) whether a component is a good clique or a bad clique. So, Reduction Rule 5.1 can be implemented in polynomial time. Similarly, by brute-forcing over all subsets of size at most $d-1$, it is possible to check whether there exists an independent set A of size at most $d-1$ that is not a Chunk. So, Reduction Rule 5.2 can be implemented in polynomial time. Reduction Rule 5.3 requires us to check whether set of vertices of one hyperedge is contained in the set of vertices of another hyperedge. This condition can be checked in polynomial time as all there are $\mathcal{O}(k^d)$ hyperedges in G. So Reduction Rule 5.3 also can be implemented in polynomial time. Constructing graph H requires us to construct S_B such that $|S_B| = \mathcal{O}(k^d)$. In order to construct Z_2, we have to construct Z_1. This can be done in polynomial time. Once Reduction Rule 5.1, 5.2 are not applicable and Z_1 is constructed, the components in F that are not in Z_1 are in Z_2. Once Z_2 is constructed, we have to check whether for every $C \in Z_2$, for every $A \in S_B$ whether $V(C) \subseteq N_G(A)$. So, H can be constructed in polynomial time. Now, Reduction Rule 5.4 can be implemented in polynomial time as this requires us to apply $(d+1)$-Expansion Lemma on a bipartite graph whose size is $\mathcal{O}(|S_B| \cdot n)$ as $|Z_2| \leq n$. This completes the proof. □

Lemma 5.6. *When Reduction Rules 5.1, 5.2, 5.3, 5.4 are not applicable, then the input graph has been transformed to an equivalent hypergraph with $\mathcal{O}(k^d)$ vertices. Each hyperedge has size at most d and hyperedges of size more than two are present only in S.*

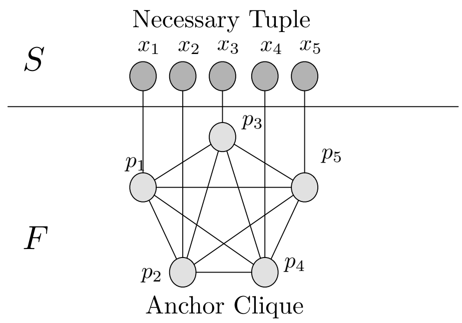

Figure 4: Illustrations of an anchor clique and a necessary tuple. In this example, $\{p_1, p_2, p_3, p_4, p_5\}$ is an *anchor clique* and $\{x_1, x_2, x_3, x_4, x_5\}$ is the corresponding *necessary tuple*.

Proof. As Reduction Rule 5.1 is not applicable, every component $C \in G[F]$ is a bad clique. So, for every bad clique is either a component in Z_1 or in Z_2. By hypothesis, precondition of Lemma 5.3 is satisfied. By Lemma 5.3, $|Z_1|$ is $\mathcal{O}(k^d)$. When Reduction Rule 5.4 is not applicable, we have that $|Z_2| \leq (d+1)|S_B| \leq (d+1) \cdot \binom{k}{d}$. So the number of components in $G[F]$ is $\mathcal{O}(k^d)$. Hence the equivalent hypergraph contains $\mathcal{O}(d \cdot k^d) = \mathcal{O}(k^d)$ vertices. $\square$

By Lemma 5.6, we have that VC-PARAM-d-CVD has a compression consisting of $\mathcal{O}(k^d)$ vertices. In particular, in that compression there are $\mathcal{O}(k^d)$ components in $G[F]$ and each component has at most d vertices. When $G[F]$ is a graph of degree at most 1, every component of $G[F]$ is either an isolated vertex or an edge. Then, the problem is equivalent to VC-PARAM-2-CVD. Then, the reduced hypergraph is a graph. In particular, this problem is equivalent to VC-1-MOD. So, we have the following corollary.

Corollary 5.1. VC-1-MOD *has a kernel with $\mathcal{O}(k^2)$ vertices and $\mathcal{O}(k^3)$ edges.*

Now, the next step is to convert our resulting hypergraph into a graph. We have added hyperedges only to capture constraints that some of the subsets of size more than two also intersects any optimal solution. In order to transform our hypergraph into a graph, we need to define the following two terms, *anchor clique* and *necessary tuple* (variations of which were defined by Fomin and Strømme [16]).

Definition 5.4 (Anchor Clique)**.** *Let C be a component of $G[F]$ with vertex set $\{p_1, \ldots, p_r\}$ where $3 \leq r \leq d$. Then, C is said to be an* anchor clique *if there exists an independent set $X = \{x_1, \ldots, x_r\} \subseteq S$ such that the following things are true.*

- *$N_G(V(C)) = X$.*
- *For every $i \in [r]$, we have that $N_G(p_i) = (C \cup \{x_i\}) \setminus \{p_i\}$.*

(see Figure 4 for an illustration)

Definition 5.5 (Necessary Tuple)**.** *Let $X \subseteq S$ be an independent set. Then, X is said to be a* necessary tuple *if there exists an anchor clique C such that $N_G(V(C)) = X$ (see Figure 4 for an illustration).*

The intuition behind defining *anchor clique* and *necessary tuple* is to identify those independent sets in S which must have a nonempty intersection with any minimum vertex cover. Therefore, we have the following lemma.

Lemma 5.7. *Suppose that G has a vertex cover of size at most ℓ and $\mathcal{S} = \{X \subseteq S | X$ is a necessary tuple $\}$, then G also has a vertex cover D of size at most ℓ such that for every $X \in \mathcal{S}, D \cap X \neq \emptyset$.*

Proof. Let G has a vertex cover D' of size at most ℓ. If for all $X \in \mathcal{S}, D' \cap X \neq \emptyset$, then we are done. So, let us assume that there exists $X \in \mathcal{S}$ such that $D' \cap X = \emptyset$. Then, consider the anchor clique C for which $N_G(V(C)) = X$. As $X \cap D' = \emptyset$ and D' is a vertex cover, we have that $V(C) \subseteq D'$. As C is an anchor clique, for any $p \in V(C)$, we have $N_G(p) \cap S = N_G(p) \cap X = \{x\}$ for some $x \in X$. Now, we construct D from D' as follows. We take some $p \in V(C)$ arbitrarily and construct D as follows. $D = (D \cup \{x\}) \setminus \{p\}$. Clearly $|D| \leq \ell$. Also D is a vertex cover of G because $N_G(p) = (V(C) \cup \{x\}) \setminus \{p\}$ and by construction $N_G(p) \subseteq D$. So, there exists a vertex cover G containing at least one vertex from every necessary tuple. □

Now, we need one more reduction rule which will convert the hypergraph into a graph in which the asymptotic number of vertices will not increase. However, we apply the following reduction rule for the first time when Reduction Rules 5.1, 5.2, 5.3, 5.4 are not applicable. Moreover, after one application of the following reduction rule, we do not apply the earlier reduction rules (i.e. Reduction Rules 5.1, 5.2, 5.3, 5.4) anymore.

Reduction Rule 5.5. *Let (G, S, ℓ) be an irreducible instance that achieves the compression provided by Lemma 5.6. Let $X \in E(G)$ be a hyperedge of size at least three. Then remove X from $E(G)$ and add an anchor clique C such that $N_G(V(C)) = X$. In particular, if $X = \{x_1, \ldots, x_r\}$, then $V(C) = \{p_1, \ldots, p_r\}$ and the followings are satisfied:*

- $V(G') \leftarrow V(G) \cup V(C)$.
- $S' \leftarrow S$.
- $E(G') \leftarrow (E(G) \cup \{(p_i, x_i) | i \in [r]\} \cup \{(p_i, p_j) | i \neq j, i, j \in [r]\}) \setminus X$.
- $\ell' \leftarrow \ell + r - 1$.

Lemma 5.8. *Reduction Rule 5.5 is safe and can be implemented in polynomial time.*

Proof. ($\Rightarrow$) Let D be a minimum vertex cover of G. Let $X \in E(G)$ be a hyperedge of size at least 3. Let C be the anchor clique added to get G' such that $N_{G'}(V(C)) = X$. As X was hyperedge in G, we know that $D \cap X \neq \emptyset$. Then, $V(C') \not\subseteq N_{G'}(X \setminus D)$. So, $|N_{G'}(X \setminus D)| < |X| = r$. So, we can add $|V(C)| - 1$ many vertices into D and get a vertex cover of G' of size at most $|D| + r - 1$.
($\Leftarrow$) Let D' be a minimum vertex cover of G'. If $X \cap D' \neq \emptyset$, then we are done. In that case we construct $D = D' \setminus V(C)$ to get a vertex cover of size at most $|D'| - r + 1$. Otherwise C be the anchor clique in G' such that $N_{G'}(V(C)) = X$. So, X is a necessary tuple in G'. By Lemma 5.7, we can construct another minimum vertex cover D'' of G' such that $X \cap D'' \neq \emptyset$. Then $|N_G(X \setminus D'') \cap V(C)| < |V(C)|$. So $|D'' \setminus V(C)| = |D''| - |V(C)| + 1 = |D'| - r + 1$. So $D = D'' \setminus V(C)$ be a vertex cover of G as D intersects the hyperedge X. So, the reduction rule is safe.
This reduction rule has to find out whether there exists a hyperedge of size at least 3 and at most d in G. Also such hyperedges can only have vertices from S. As d is a constant, this reduction rule can be implemented in polynomial time. □

We keep applying Reduction Rule 5.5 as long as there exists a hyperedge with more than 2 vertices in the graph. Now, we have the following theorem.

Theorem 5.1. VC-PARAM-d-CVD *has a kernel with $\mathcal{O}(k^d)$ vertices.*

Proof. Our kernelization algorithm goes as follows. We first apply Lemma 5.6 on the input instance (G, S, ℓ) where $|S|$ is the parameter. Initial input instance is a graph. Applying Lemma 5.6, we get an instance for vertex cover (also called a hitting set) in a hypergraph (G', S', ℓ') where $|V(G')| = \mathcal{O}(k^d)$ and $|S'| \leq k$. In particular, all hyperedges of size at least three and at most d are completely contained in S'. So, $G' \setminus S'$ is a graph. Moreover, there are $\mathcal{O}(k^d)$ components in $G' \setminus S'$. Next, we apply only Reduction Rule 5.5 in the instance (G', S', ℓ') as long as there remains a hyperedge of size at least three and at most d in $G' \setminus S'$. We get the instance (G'', S'', ℓ'') where $S'' = S'$. Note that by construction, at most $\sum_{i=3}^{d} \binom{k}{i} \leq d \cdot k^d$ anchor cliques are added to $G' \setminus S'$ to get G''. So, number of components in $G'' \setminus S''$ is still $\mathcal{O}(k^d)$. As each component in G'' that was not in G' has at most d vertices, we get a graph with $\mathcal{O}(k^d)$ vertices. So, VC-PARAM-d-CVD has a kernel with $\mathcal{O}(k^d)$ vertices. □

6 Kernel Lower Bounds

Here we prove lower bounds, under complexity theoretic assumptions, for the size of the kernel of the problems we considered in this paper. We prove these by giving a *parameter preserving transformation* from d-CNF-SAT to our problem(s) and use a theorem due to Dell and van Melkebeek [9]. Here d-CNF-SAT is the problem of testing the satisfiability of a d-CNF formula, a boolean formula where the clauses are in CNF form with at most d literals each. We use the terminologies of *oracle communication protocol and parameter preserving transformation* for this. The definitions and the known results we use are given below.

Definition 6.1 (Oracle Communication Protocol). *(See [11, 9]) Let $L \subseteq \Sigma^*$ be a language. An oracle communication protocol for language L is a communication protocol with two players* Alice *and* Bob. *Alice is given an input $x \in \Sigma^*$ and can only use poly($|x|$) time for her computations. Player Bob is computationally unbounded, but not given any part of x. At the end of the protocol, Alice should be able to decide if $x \in L$ using help (communication) from Bob. The cost of the protocol is the number of bits of communication between Alice and Bob.*

We know the following theorem due to Dell and van Melkebeek [9].

Theorem 6.1 (Lower Bound for d-CNF-SAT [9]). *The d-CNF-SAT has no oracle communication protocol of cost $\mathcal{O}(n^{d-\varepsilon})$ for any $d \geq 3, \varepsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.*

The above result by Dell and van Melkebeek can be rephrased as follows (see [7, 24]).

Theorem 6.2 ([7, 24]). *Let ϕ be an instance of d-CNF-SAT with n variables and m clauses for some $d \geq 3$. If there exists a polynomial time algorithm that transforms ϕ to an equivalent instance of an arbitrary problem with $\mathcal{O}(n^{d-\varepsilon})$ bits for some $\varepsilon > 0$, then $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.*

Note that we use the *size* of a graph to mean the *number of vertices and edges* of that graph in the following theorems. Before that, we need the notion of *parameter preserving transformation.*

Definition 6.2 (Parameter preserving transformation (PPT)). *Let Π_1 and Π_2 be two parameterized problems. We say that there exists a* parameter preserving transformation *from Π_1 to Π_2 if there exists a polynomial time algorithm $\mathcal{B}$ that given an instance (x, k) of Π_1, constructs an instance (x', k') of Π_2 such that*

- *$(x, k) \in \Pi_1$ if and only if $(x', k') \in \Pi_2$, and*
- *$k' \leq ck$ for some constant c independent of $|x|$ and k.*

There is a standard folklore reduction (see for example [33], when $d = 3$) from d-CNF-SAT to VC-PARAM-d-CVD. We observe that this reduction is actually a parameter preserving transformation and provide here for completeness.

Theorem 6.3. *There exists a parameter preserving transformation from the d-CNF-SAT parameterized by the number of variables to VC-PARAM-d-CVD. In VC-PARAM-d-CVD, the size of the modulator is twice the number of variables in the d-CNF-SAT formula.*

Proof. Let ϕ be the given d-CNF formula having n variables and m clauses. Every clause has at most d literals. In the graph we construct, corresponding to each variable x, we will have an edge $(x, \overline{x})$. Corresponding to a clause C_j with d_j literals, we will have a clique of size d_j with vertices labeled with the corresponding literals in the clause. More specifically, let $x_1, x_2, \ldots, x_n$ be the variables, and let C_j be a clause. We name a vertex of the clique corresponding to the clause to be $y_{a,j}$ if x_a is a literal in C_j and $\overline{y_{a,j}}$ if $\overline{x_a}$ is a literal in C_j. Let $S = \{x_i | i \in [n]\} \cup \{\overline{x_i} | i \in [n]\}$ be the modulator. For every $i \in [n], j \in [m]$, we draw edges $(x_i, y_{i,j})$ and $(\overline{x_i}, \overline{y_{i,j}})$. As $|S| = 2n$ and this transformation can be performed in polynomial time, the theorem follows from the claim that the given formula ϕ is satisfiable if and only if G has a vertex cover of size at most $n + \sum_{j=1}^{m} (d_j - 1)$.

Note that any vertex cover of G has size at least $n + \sum_{j=1}^{m}(d_j - 1)$ as we have a matching of size n in S and the edges of the j'th clique (corresponding to clause C_j) of F requires $(d_j - 1)$ vertices to cover. If ϕ is satisfiable, then we find the vertex cover of G as follows. We pick the vertices of S according to the assignment that satisfies ϕ (i.e. for each variable x, pick x if x is set to be true in the satisfying assignment and pick $\overline{x}$ otherwise.) And each clause C_j has a literal, say, x_a that makes the clause satisfiable, so we pick to the vertex cover, all but the $y_{a,j}$ vertex from the corresponding clique. Edges between $y_{a,j}$ and S are covered by x_a that would have been picked in the vertex cover.

Conversely, let R be a vertex cover of G of size $\sum_{j=1}^{m}(d_j - 1) + n$. Consider $R \cap S$. As S has an edge between x_i and $\overline{x_i}$ for every i, exactly one from x_i and $\overline{x_i}$ is present in R. Now, we construct an assignment $\overline{b}$ from $R \cap S$. If $x_i \in R \cap S$, then set $x_i = \mathsf{true}$ and set $x_i = \mathsf{false}$ otherwise. The claim is that every clause is satisfiable. This is because for the clique corresponding to the clause C_j, exactly $d_j - 1$ vertices are in R, and hence some vertex, say $y_{a,j}$ is not in R. Then to cover the edge $(y_{a,j}, x_a)$, x_a must be in $R \cap S$ which means that the literal x_a makes the clause C_j satisfiable. Thus each clause of ϕ is satisfied by the assignment. $\square$

Theorem 6.4. *For any $d \geq 3, \varepsilon > 0$,* VC-PARAM-$d$-CVD *has no kernel of size $\mathcal{O}(k^{d-\varepsilon})$ unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. *In particular, for any $d \geq 3, \varepsilon > 0$, there exists no polynomial time algorithm that transforms a given instance of* VC-PARAM-d-CVD *to an equivalent instance of any arbitrary problem with $\mathcal{O}(k^{d-\varepsilon})$ bits unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

Proof. Let (G, S, ℓ) be an instance of VC-PARAM-d-CVD. Suppose that there exists a polynomial time algorithm $\mathcal{B}$ that transforms (G, S, ℓ) to an equivalent instance $x \in \Sigma^*$ of an arbitrary problem $L \subseteq \Sigma^*$ such that $|x|$ can be represented using $\mathcal{O}(|S|^{d-\varepsilon})$ bits. Then, given an instance ϕ of d-CNF-SAT with n variables and m clauses, first we apply the polynomial time algorithm given by Theorem 6.3 and get an instance (G, S, ℓ) of VC-PARAM-d-CVD such that $|S| = 2n$. Now, we apply the algorithm $\mathcal{B}$ that transforms (G, S, ℓ) to an equivalent instance $x \in \Sigma^*$ of $L \subseteq \Sigma^*$. We know by our assumption that x can be represented using $\mathcal{O}(|S|^{d-\varepsilon})$ bits, i.e. $\mathcal{O}(n^{d-\varepsilon})$ bits. But this will imply from Theorem 6.2 that $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. Now, if VC-PARAM-$d$-CVD has a kernel of size $\mathcal{O}(k^{d-\varepsilon})$, then this kernel has $\mathcal{O}(k^{d-\varepsilon})$ vertices and edges. The adjacency list of that graph takes $\mathcal{O}(k^{d-\varepsilon}\log_2(k^{d-\varepsilon}))$ bits to represent this graph. Hence that graph can be represented using $\mathcal{O}(k^{d-\varepsilon'})$ bits for some $\varepsilon' > 0$. Then, this kernel will provide a polynomial time algorithm that transforms an instance of VC-PARAM-d-CVD to an instance of VERTEX COVER problem with $\mathcal{O}(k^{d-\varepsilon'})$ bits for some $\varepsilon' > 0$. This will also imply $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. This completes the proof of the theorem. $\square$

Note that a collection of cliques of size at most three is a subclass of graphs with degree at most two. So we have that VC-PARAM-3-CVD is a special case of VC-2-MOD. So, the following corollary follows from Theorem 6.4 when $d = 3$.

Corollary 6.1. VC-PARAM-3-CVD *has no kernel of size $\mathcal{O}(k^{3-\varepsilon})$ for any $\varepsilon > 0$ unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. *In particular,* VC-2-MOD *has no kernel of size $\mathcal{O}(k^{3-\varepsilon})$ for any $\varepsilon > 0$ unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

7 Conclusion

In this paper we gave polynomial kernels for VC-2-MOD and other related problems. There is a gap between upper and lower bounds on the kernel sizes we obtained for VC-2-MOD. It would be interesting to bridge this gap. We also know that VERTEX COVER admits an $\mathcal{O}(k^{12})$ vertex kernel when parameterized by the deletion distance to a pseudo-forest [17]. It would be interesting to see whether the ideas for VC-2-MOD can be extended to get an improved kernel for this problem. It is known that VERTEX COVER admits a randomized polynomial kernel parameterized by the odd cycle transversal number of the graph (minimum number of vertices whose deletion results in a bipartite graph). Is it possible to obtain a deterministic polynomial kernel for this parameterization (maybe using some ideas from this paper)? We think this might be easier and probably the first step towards obtaining a deterministic polynomial kernel for the ODD CYCLE TRANSVERSAL problem itself which is a long standing open problem.

References

[1] Faisal N. Abu-Khzam. A Kernelization Algorithm for d-Hitting Set. *J. Comput. Syst. Sci.*, 76(7):524–531, 2010.

[2] Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernelization Lower Bounds by Cross-Composition. *SIAM J. Discrete Math.*, 28(1):277–305, 2014.

[3] Marin Bougeret and Ignasi Sau. How much does a treedepth modulator help to obtain polynomial kernels beyond sparse graphs? *Proceedings of IPEC 2017*, abs/1609.08095, 2017.

[4] Leizhen Cai. Parameterized Complexity of Vertex Colouring. *Discrete Applied Mathematics*, 127(3):415–429, 2003.

[5] Jianer Chen, Iyad A. Kanj, and Ge Xia. Improved upper bounds for vertex cover. *Theor. Comput. Sci.*, 411(40-42):3736–3756, 2010.

[6] Robert Crowston, Michael R. Fellows, Gregory Gutin, Mark Jones, Eun Jung Kim, Fran Rosamond, Imre Z. Ruzsa, Stéphan Thomassé, and Anders Yeo. Satisfying more than half of a system of linear equations over GF(2): A multivariate approach. *J. Comput. Syst. Sci.*, 80(4):687–696, 2014.

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

[8] Marek Cygan, Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. On the Hardness of Losing Width. *Theory Comput. Syst.*, 54(1):73–82, 2014.

[9] Holger Dell and Dieter van Melkebeek. Satisfiability Allows No Nontrivial Sparsification unless the Polynomial-Time Hierarchy Collapses. *J. ACM*, 61(4):23:1–23:27, 2014.

[10] Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

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

[12] Michael Etscheid and Matthias Mnich. Linear Kernels and Linear-Time Algorithms for Finding Large Cuts. *Algorithmica*, 2017.

[13] Michael R. Fellows, Bart M. P. Jansen, and Frances A. Rosamond. Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity. *Eur. J. Comb.*, 34(3):541–566, 2013.

[14] Michael R. Fellows, Daniel Lokshtanov, Neeldhara Misra, Matthias Mnich, Frances A. Rosamond, and Saket Saurabh. The Complexity Ecology of Parameters: An Illustration Using Bounded Max Leaf Number. *Theory Comput. Syst.*, 45(4):822–848, 2009.

[15] Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, Geevarghese Philip, and Saket Saurabh. Hitting Forbidden Minors: Approximation and Kernelization. *SIAM J. Discrete Math.*, 30(1):383–410, 2016.

[16] Fedor V. Fomin and Torstein J. F. Strømme. Vertex Cover Structural Parameterization Revisited. *CoRR*, abs/1508.00395, 2016.

[17] Fedor V. Fomin and Torstein J. F. Strømme. Vertex Cover Structural Parameterization Revisited. In *Graph-Theoretic Concepts in Computer Science - 42nd International Workshop, WG 2016, Istanbul, Turkey, June 22-24, 2016, Revised Selected Papers*, pages 171–182, 2016.

[18] Martin Grötschel and George L. Nemhauser. A polynomial algorithm for the max-cut problem on graphs without long odd cycles. *Math. Programming*, 29(1):28–40, 1984.

[19] Gregory Gutin, Eun Jung Kim, Stefan Szeider, and Anders Yeo. A probabilistic approach to problems parameterized above or below tight bounds. *J. Comput. Syst. Sci.*, 77(2):422–429, 2011.

[20] Gregory Gutin and Anders Yeo. Constraint Satisfaction Problems Parameterized above or below Tight Bounds: A Survey. In *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *Lecture Notes in Computer Science*, pages 257–286. Springer, 2012.

[21] Eva-Maria C. Hols and Stefan Kratsch. Smaller parameters for vertex cover kernelization. *Proceedings of IPEC 2017*, abs/1711.04604, 2017.

[22] Wen Lian Hsu, Yoshiro Ikura, and George L. Nemhauser. A polynomial algorithm for maximum weighted vertex packings on graphs without long odd cycles. *Math. Programming*, 20(2):225–232, 1981.

[23] Bart M. P. Jansen and Hans L. Bodlaender. Vertex Cover Kernelization Revisited - Upper and Lower Bounds for a Refined Parameter. *Theory Comput. Syst.*, 53(2):263–299, 2013.

[24] Bart M. P. Jansen and Astrid Pieterse. Sparsification Upper and Lower Bounds for Graph Problems and Not-All-Equal SAT. *Algorithmica*, 79(1):3–28, 2017.

[25] Eun Jung Kim and Ryan Williams. Improved parameterized algorithms for above average constraint satisfaction. In *Parameterized and Exact Computation - 6th International Symposium, IPEC 2011, Saarbrücken, Germany, September 6-8, 2011. Revised Selected Papers*, pages 118–131, 2011.

[26] Stefan Kratsch. A Randomized Polynomial Kernelization for Vertex Cover with a Smaller Parameter. In *24th Annual European Symposium on Algorithms, ESA 2016, August 22-24, 2016, Aarhus, Denmark*, pages 59:1–59:17, 2016.

[27] Stefan Kratsch and Magnus Wahlström. Representative Sets and Irrelevant Vertices: New Tools for Kernelization. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 450–459, 2012.

[28] Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Faster Parameterized Algorithms Using Linear Programming. *ACM Transactions on Algorithms*, 11(2):15:1–15:31, 2014.

[29] Diptapriyo Majumdar, Venkatesh Raman, and Saket Saurabh. Kernels for Structural Parameterization of Vertex Cover: case of small degree modulators. In *10th International Symposium on Parameterized and Exact Computation (IPEC)*, volume LIPICS: Leibniz International Proceedings in Informatics (43), pages 331–342, 2015.

[30] George L. Nemhauser and Leslie E. Trotter Jr. Vertex Packings: Structural properties and Algorithms. *Math. Program.*, 8(1):232–248, 1975.

[31] Fahad Panolan and Ashutosh Rai. On the Kernelization Complexity of Problems on Graphs without Long Odd Cycles. In *COCOON 2012*, volume 7434 of *LNCS*, pages 445–457. Springer, 2012.

[32] Elena Prieto. *Systematic Kernelization in FPT Algorithm Design.* PhD thesis, The University of Newcastle, Australia, 2005.

[33] Michael Sipser. *Introduction to the Theory of Computation.* PWS Publishing Company, 1997.

[34] Torstein J. F. Strømme. Kernelization of Vertex Cover by Structural Parameters. Master's thesis, University of Bergen, Norway, 2015.

[35] Stéphan Thomassé. A $4k^2$ kernel for feedback vertex set. *ACM Transactions on Algorithms*, 6(2), 2010.