Representative sets and irrelevant vertices: New tools for kernelization*

Stefan Kratsch[†] Magnus Wahlström[‡]

February 20, 2020

Abstract

We continue the development of matroid-based techniques for kernelization, initiated by the present authors (TALG 2014). We significantly extend the usefulness of matroid theory in kernelization by showing applications of a result on *representative sets* due to Lovász (Combinatorial Surveys 1977) and Marx (TCS 2009). As a first result, we show how representative sets can be used to derive a polynomial kernel for the elusive ALMOST 2-SAT problem (where the task is to remove at most k clauses to make a 2-CNF formula satisfiable), solving a major open problem in kernelization. This result also yields a new $\mathcal{O}(\sqrt{\log \mathrm{OPT}})$-approximation for the problem, improving on the $\mathcal{O}(\sqrt{\log n})$-approximation of Agarwal et al. (STOC 2005) and an implicit $\mathcal{O}(\log \mathrm{OPT})$-approximation due to Even et al. (JACM 2000).

We further apply the representative sets tool to the problem of finding *irrelevant vertices* in graph cut problems, that is, vertices that can be made undeletable without affecting the answer to the problem. This gives the first significant progress towards a polynomial kernel for the MULTIWAY CUT problem; in particular, we get a kernel of $\mathcal{O}(k^{s+1})$ vertices for MULTIWAY CUT instances with at most s terminals. Both these kernelization results have significant spin-off effects, producing the first polynomial kernels for a range of related problems.

More generally, the irrelevant vertex results have implications for covering min cuts in graphs. For a directed graph $G = (V, E)$ and sets $S, T \subseteq V$, let r be the size of a minimum (S, T)-vertex cut (which may intersect S and T). We can find a set $Z \subseteq V$ of size $\mathcal{O}(|S| \cdot |T| \cdot r)$ that contains a minimum (A, B)-vertex cut for every $A \subseteq S, B \subseteq T$. Similarly, for an undirected graph $G = (V, E)$, a set of terminals $X \subseteq V$, and a constant s, we can find a set $Z \subseteq V$ of size $\mathcal{O}(|X|^{s+1})$ that contains a minimum multiway cut for every partition of X into at most s pairwise disjoint subsets. Both results are polynomial time. We expect this to have further applications; in particular, we get direct, reduction rule-based kernelizations for all problems above, in contrast to the indirect compression-based kernel previously given for ODD CYCLE TRANSVERSAL (TALG 2014).

All our results are randomized, with failure probabilities that can be made exponentially small in n, due to needing a *representation* of a matroid to apply the representative sets tool.

1 Introduction

Polynomial kernelization is a formalization of the notion of polynomial-time preprocessing, or more generally of polynomial-time instance simplification and data reduction. Such reduction steps are commonly applied in practice, see, e.g., the well-known CPLEX integer programming package, or many state-of-the-art SAT solvers. However, to study this theoretically, one needs a notion of the *hardness* of an instance beyond the instance size, e.g., the length of a certificate [38] or a more generic *parameter* associated with the input

*An extended abstract of this work appeared in 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012 [46]. This version contains some additional material, including the material in Section 4, as well as a new approximation result and improved kernel size for ALMOST 2-SAT and an improved kernel size for GROUP FEEDBACK VERTEX SET.

[†]Utrecht University, kratsch@informatik.hu-berlin.de

[‡]Royal Holloway, University of London, Magnus.Wahlstrom@rhul.ac.uk

(cf. [21, 14]). Informally, a kernelization is a polynomial-time reduction of an input instance, with parameter value k, to an equivalent instance of the same problem, the *kernel*, with total output size bounded as a function of k; a problem has a *polynomial kernel* if the size bound is polynomial in k. This turns out to be a robust and interesting notion, and there is much work on both upper and lower bounds for the existence of, or best possible size of, a polynomial kernel for various problems; see [7, 29] and [6, 31, 19]. A recent breakthrough was achieved by Drucker [22], who proved, among other things, that the AND-distillation conjecture of Bodlaender et al. [6] holds assuming that NP does not admit non-uniform statistical zero-knowledge proofs (a weaker assumption than the usual $\mathsf{NP} \nsubseteq \mathsf{coNP/poly}$).

Prior to this work (that is, prior to the extended abstract in 2012 [46]), there were two major groups of problems for which the existence of polynomial kernels remained open. The first group is centered around the ALMOST 2-SAT problem: Given a 2-CNF formula $\mathcal{F}$ and an integer k, can one remove at most k clauses to make $\mathcal{F}$ satisfiable (or, equivalently, find an assignment under which at most k clauses are not satisfied)? This is a natural, expressive problem which (at least for purposes of parameterized complexity and kernelization) captures several problems of independent interest. For one thing, it directly expresses ODD CYCLE TRANSVERSAL (OCT); the existence of a polynomial kernel for OCT was a long-standing open problem, until being solved by the present authors [47]. Less directly, a polynomial kernel for ALMOST 2-SAT has been shown to imply the same for VERTEX COVER ABOVE MATCHING, KŐNIG VERTEX DELETION for graphs with perfect matchings, and the RHORN-BACKDOOR DELETION SET problem from practical SAT solving (cf. [20, 44]), among other problems; see [66, 33, 61]. We add to the list VERTEX COVER ABOVE LP, i.e., VERTEX COVER parameterized by the size of the LP gap. For all of these problems, no polynomial kernel was previously known.

The second group of open problems represents the class of *graph cut* problems. This is a wide class, where little is known regarding polynomial kernelization; problems for which polynomial kernelization is open include DIRECTED FEEDBACK VERTEX SET (arguably one of the biggest open problems in kernelization; see [8]), MULTIWAY CUT, and MULTICUT under various parameterizations, as well as GROUP FEEDBACK ARC/VERTEX SET, which again generalizes OCT.

In this paper, we show polynomial kernels for ALMOST 2-SAT, and for a collection of graph cut problems, including MULTIWAY CUT with a constant number of terminals and MULTICUT with a constant number of cut requests. We also show results about covering min cuts and multiway cuts through a set of terminals using few vertices, which should be of independent interest. We make use of a lemma on *representative sets* from matroid theory, due to Lovász [51] and Marx [53]. In particular, we show how to apply the lemma in *irrelevant vertex* arguments, i.e., how to use it to find vertices in cut problems which can be made undeletable without affecting the outcome. All our results are randomized, with failure probabilities which can be made exponentially small in the input size.

1.1 Our Techniques and Results

An extended illustration of the kernelization approach used in this paper is given in Section 4; below, we give a brief overview of the technical tools.

The most important result used in this paper is the *representative sets lemma* due to Lovász [51] and Marx [53]. To state this, we need to recall a few definitions (see also Sections 2 and 3).

A *matroid* is a pair $M = (E, \mathcal{I})$ where $\mathcal{I} \subseteq 2^E$, referred to as the *independent sets* of M, is a set of subsets of E subject to certain axioms (see Section 2). A particularly important case is a *linear matroid*. Let A be an $m \times n$ matrix over a field $\mathbb{F}$, and let E be the set of columns of A. Then A defines a matroid $M = (E, \mathcal{I})$, where $\mathcal{I}$ contains the sets of column vectors that are linearly independent over $\mathbb{F}$, and we say that A *represents* M. A matroid is linear if it is representable in this way.

For a set $X \subseteq E$, we say that a set $Y \subseteq E$ *extends* X if $X \cap Y = \emptyset$ and $X \cup Y \in \mathcal{I}$; this requires that both X and Y are independent. The representative sets lemma shows that for each collection $\mathcal{Y}$, there is a subcollection $\mathcal{Y}^*$ of bounded size that extends exactly the same sets X; we will later say that $\mathcal{Y}^*$ *represents* $\mathcal{Y}$.

Lemma 1 (representative sets lemma [51, 53])**.** *Let $M = (E, \mathcal{I})$ be a linear matroid represented by a matrix*

A of rank $r+s$, and let $\mathcal{Y}$ be a collection of independent sets of M, each of size s. Assume that s is a constant. Then in polynomial time in size of A and size of $\mathcal{Y}$ we can compute a set $\mathcal{Y}^* \subseteq \mathcal{Y}$ of size at most $\binom{r+s}{s}$ such that for every set $X \subseteq E$, there is a set in $\mathcal{Y}$ that extends X if and only if there is a set in $\mathcal{Y}^*$ that extends X.

We show several powerful applications of this lemma in polynomial kernelization.

The power of the lemma lies in that the set X does not need to be provided; the algorithm only takes the matrix A and the family $\mathcal{Y}$ as input, yet the output family $\mathcal{Y}^*$ works for any set X. Let us review a quick example (see Section 4 for more details). Let $(G=(V,E),k)$ be an instance of Vertex Cover, and let M be the uniform matroid over V of rank $k+2$. Apply the lemma to M, with $\mathcal{Y}=E$, and let $E' \subseteq E$ be the edges retained in the output $\mathcal{Y}^*$. Then it can be verified that for every $X \subseteq V$, $|X| \leq k$, the set X is a vertex cover of G if and only if there is no edge $Y \in \mathcal{Y}^*$ that extends X. Hence (G,k) is a positive instance of Vertex Cover if and only if $(G'=G(E'),k)$ is, and we have reduced the input to a graph of $|E'| = \mathcal{O}(k^2)$ edges (the kernel).

Still, although the above lemma is powerful, it is also non-trivial to apply to any given kernelization task, and the main work of this paper consists of identifying applications – i.e., matroids M and families $\mathcal{Y}$ – that are useful for kernelization, and reducing the task of kernelizing a problem to a situation where such a result applies.

The most powerful applications come from *gammoids*, which are a family of representable matroids encoding cuts in directed graphs (see Section 2). Crucially, while every vertex of a graph may participate in *some* (A,B)-min cut for fixed (A,B), we are able to capture the property of a vertex v to occur in *all* (A,B)-min cuts as the question of extending a certain independent set $X(A,B)$, depending only on A and B, by an independent set $Y(v)$, depending only on v, in a specially created gammoid M. Moreover, this can be done in such a way so that it can be applied simultaneously to all sets $A \subseteq S$ and $B \subseteq T$ for given vertex sets $S,T \subseteq V$.

Using this together with the representative sets lemma, in polynomial time we are able to identify a set $Z \subseteq V$ of $\mathcal{O}((|S|+|T|)^3)$ vertices that contains all such "essential" or bottleneck cut vertices v for every one out of $2^{|S|+|T|}$ cut problems created by selecting $A \subseteq S$ and $B \subseteq T$. Variants of this, combined with an irrelevant vertex approach, allow us to compute polynomial-sized *cut-covering sets* for various notions of terminal cuts (see Theorem 1 and its corollaries).

The above ideas allow us to produce a range of kernelization results for Almost 2-SAT and related problems; see below. We also consider a multi-partite version of the above setting (for details, see Section 5), which has applications both to problems such as Multiway Cut and Multicut and to a highly general problem called Group Feedback Vertex Set.

Our results: Cut-covering sets. Our possibly most surprising results have to do with covering min cuts and multiway cuts in graphs. We have the following results (note that all our vertex cuts are allowed to overlap the terminal set).

Theorem 1. *Let $G=(V,E)$ be a directed graph and let $S,T \subseteq V$. Let r denote the size of a minimum (S,T)-vertex cut (which may intersect S and T). There exists a set $Z \subseteq V$, $|Z| = \mathcal{O}(|S| \cdot |T| \cdot r)$, such that for any $A \subseteq S$ and $B \subseteq T$, it holds that Z contains a minimum (A,B)-vertex cut. We can find such a set in randomized polynomial time with failure probability $\mathcal{O}(2^{-|V|})$.*

Applying this result to an auxiliary graph construction gives us the following, more expressive version.

Corollary 1. *Let $G=(V,E)$ be a directed graph, and $W \subseteq V$ a set of terminals. We can identify, in randomized polynomial time, a set $Z \supseteq W$ of $\mathcal{O}(|W|^3)$ vertices such that for any partition $W = A \cup B \cup R \cup Q$, a minimum (A,B)-vertex cut in $G-R$ is contained in Z.*

For undirected graphs, we also show corresponding variants for multi-way partitions into constantly many parts.

Theorem 2. *Let $G = (V, E)$ be an undirected graph and $X \subseteq V$. For any s, there exists a set $Z \subseteq V$, $|Z| = \mathcal{O}(|X|^{s+1})$, such that for any partition $\mathcal{X} = (X_1, \ldots, X_s)$ into exactly s parts, the set Z contains a minimum multiway cut of $\mathcal{X}$ (i.e., a minimum cut C such that no pairs of distinct sets X_i, X_j are connected to each other in $G - C$). We can find such a set in randomized polynomial time with failure probability $\mathcal{O}(2^{-|V|})$.*

Corollary 2. *Let $G = (V, E)$ be an undirected graph and $X \subseteq V$. Let s be a constant. Then there is a set $Z \subseteq V$ of $\mathcal{O}(|X|^{s+1})$ vertices such that for every partition $\mathcal{X} = (X_0, X_X, X_1, \ldots, X_s)$ of X into $s+2$ possibly empty parts, the set Z contains a minimum multiway cut of $(X_1, \ldots, X_s)$ in the graph $G - X_X$. We can compute such a set in randomized polynomial time with failure probability $\mathcal{O}(2^{-|V|})$.*

Almost 2-SAT and related problems. Applying the above methods and results to kernelization problems, we get several new results. We first focus on ALMOST 2-SAT. We give a kernel for this problem in several steps. We begin by studying the compression form of the problem, where the input in addition to an ALMOST 2-SAT instance $(\mathcal{F}, k)$ also contains a suboptimal solution $X \subseteq V(\mathcal{F})$. We capture the essence of this problem in a constrained graph cut problem we call DIGRAPH PAIR CUT, and show a polynomial kernelization for this problem using the representative sets lemma. In DIGRAPH PAIR CUT we are given a directed graph $G = (V, E)$, a source vertex s, a set of pairs $\mathcal{P} \subseteq \binom{V}{2}$, and an integer k; the task is to delete a set X of at most k vertices other than s such that for each $\{u, v\} \in \mathcal{P}$ at least one of u and v is not reachable from s in $G - X$. We get the following.

Theorem 3. DIGRAPH PAIR CUT *admits a randomized polynomial kernelization with $\mathcal{O}(k^4)$ vertices, where k is the bound on the solution size. The error is 2-sided, with error probability $\mathcal{O}(2^{-n})$.*

Using this, from an instance $(\mathcal{F}, k)$ of ALMOST 2-SAT and a solution X to $\mathcal{F}$, we can produce a kernel for $(\mathcal{F}, k)$ of $\mathcal{O}(|X|^4)$ variables. To complete the kernelization, we need to produce a small initial solution X. In the previous version of this paper [46], we used the $\mathcal{O}(\sqrt{\log n})$-approximation of Agarwal et al. [3] in combination with an FPT algorithm to produce a solution X with $|X| = \mathcal{O}(k^{1.5})$, resulting in a kernel of $\mathcal{O}(k^6)$ vertices.[1] Here, we show that the above kernelization can be adapted to preserve solution sizes. Thus, by applying the approximation algorithm to the output of the previous kernelization, we are able to produce a better approximation and a smaller kernel.

Theorem 4. ALMOST 2-SAT *admits a randomized polynomial-time $\mathcal{O}(\sqrt{\log OPT})$-approximation.*

Theorem 5. ALMOST 2-SAT *with a bound k on the solution size has a randomized kernel with $\mathcal{O}(k^4 \log^2 k)$ variables and failure probability $\mathcal{O}(2^{-n})$, limited to false negatives.*

As mentioned above, this gives the first polynomial kernels for a range of problems.

Corollary 3. *The following problems all have randomized polynomial kernels:* VERTEX COVER ABOVE MATCHING, VERTEX COVER PARAMETERIZED BY KŐNIG DELETION SET, KŐNIG VERTEX DELETION ON GRAPHS WITH PERFECT MATCHING, RHORN-BACKDOOR DELETION SET, *and* VERTEX COVER ABOVE LP.

Multiway Cut and Further Applications. Finally, we move on to kernelization results for cut problems defined via multiple terminal sets or cut requests.

Theorem 6. *The following kernelizations are possible:* MULTIWAY CUT WITH DELETABLE TERMINALS(k), *with $\mathcal{O}(k^3)$ vertices; s-*MULTIWAY CUT(k), *with $\mathcal{O}(k^{s+1})$ vertices; s-*MULTICUT(k), *with $\mathcal{O}(k^{\lceil\sqrt{2s}\rceil+1})$ vertices;* GROUP FEEDBACK VERTEX SET(k), *for a group of s elements, with $\mathcal{O}((k \log k \log\log k)^{s+1})$ vertices. All results are randomized, with failure probability exponentially small in n. The errors are limited to false negatives.*

[1] We have since been made aware of a result of Even et al. regarding SYMMETRIC MULTICUT in directed graphs [24], which implies a $\mathcal{O}(\log \mathrm{OPT} \log\log \mathrm{OPT})$-approximation for ALMOST 2-SAT. However, the $\mathcal{O}(\sqrt{\log \mathrm{OPT}})$-approximation we show is not previously known.

1.2 Related work

ALMOST 2-SAT (also known as MIN 2CNF DELETION) was shown to be FPT, runtime $\mathcal{O}^*(15^k)$,[2] by Razgon and O'Sullivan [67]; this has been improved to $\mathcal{O}^*(9^k)$ [66], $\mathcal{O}^*(4^k)$ [17], and $\mathcal{O}^*(2.6181^k)$ [61]. It has an $\mathcal{O}(\sqrt{\log n})$-approximation by Agarwal et al. [3], an $\mathcal{O}(\log \mathrm{OPT} \log\log \mathrm{OPT})$-approximation due to Even et al. (via SYMMETRIC MULTICUT) [24], and no constant factor approximation under the Unique Games Conjecture [43].

Graph cut problems have been a catalyst for the development of new techniques in parameterized complexity, including the now ubiquitous *iterative compression* technique [68, 35], the notion of *important separators* [52], the *shadow removal* technique [54], and the approach of *recursive understanding* [11]. Our focus here is on MULTIWAY CUT(k), first shown to be FPT by Marx [52]. The currently fastest algorithm [17], runtime $\mathcal{O}^*(2^k)$, uses an LP approach based on work of Guillemot [34]; we also use some insights of the latter.

As for polynomial kernelization of graph cut problems, in joint work with Cygan, Pilipczuk, and Pilipczuk [15] the present authors show, amongst others, that MULTICUT(k) and DIRECTED 2-MULTIWAY CUT(k) do not admit polynomial kernels unless the polynomial hierarchy collapses. Apart from this, and previous work [47] for ODD CYCLE TRANSVERSAL, little is known about kernels for cut or feedback problems beyond the kernelizations for FEEDBACK VERTEX SET, e.g., [71].

For a long time (e.g., up until the publication of the extended abstract [46]), matroids saw little use in parameterized algorithms, either as tools or as a problem setting. Pioneering work includes Marx [53] and Marx and Schlotter [55]. However, since the publication of [46], matroid-based tools have become an important ingredient, especially in the construction of fast deterministic FPT algorithms [28, 14]. Consequently, there has also been interest in derandomizing common matroid operations and representation constructions [49, 59]. The core challenge of a deterministic representation for transversal matroids and gammoids remains unsolved, although there have been recent breakthroughs on related problems [26, 36]. Linear representations in general have also been used for kernelization and other tasks of compact information representation [47, 72, 4, 56].

Irrelevant vertex arguments are a central part of the DISJOINT PATHS algorithm of Robertson and Seymour [69], which lies behind the celebrated FPT algorithm for testing graph minors. However, the arguments used by Robertson and Seymour to locate irrelevant vertices are very different from those used in this paper (and the resulting bounds are far from polynomial).

Moitra [60] defined and constructed vertex cut sparsifiers, which, given a graph G and a set of terminals X, approximate the values of all terminal cuts in G using (capacitated) edges on vertex set X only. This has led to a sequence of follow-up work; closest to our setting is Chuzhoy [13], who gives a constant-factor approximation result using $\mathcal{O}(C^3)$ vertices, where C is the total capacity of the terminals X (assuming every edge has capacity at least one). The present work differs from hers in that, on the one hand, we do not consider weighted edges; on the other hand, our constructions are exact, run in polynomial time in both n and k, and also cover directed graphs (and vertex deletions); see Theorem 1 and Corollary 1. Chuzhoy also covers the more general case of flow sparsifiers (see [48]).

Follow-up work. Several recent papers have applied the techniques introduced in this work (that is, in the extended abstract [46]). Fafianie et al. [25] use the two-way cut-covering result for a preprocessing under uncertainty result for BIPARTITE MATCHING. Hols and Kratsch [40] obtain a randomized polynomial kernelization for the SUBSET FEEDBACK VERTEX SET problem where one needs to delete at most k vertices to remove all cycles that contain at least one edge or vertex from a specified set; the key argument in the kernelization is a more involved version of what is used for the MULTIWAY CUT WITH DELETABLE TERMINALS(k) (see Section 4.2), which is generalized by this problem. Kratsch [45] gave a randomized polynomial kernelization for VERTEX COVER PARAMETERIZED ABOVE LOVÁSZ-PLUMMER BOUND, complementing an earlier FPT-algorithm by Garg and Philip [32]. The kernelization uses the approach taken for DIGRAPH PAIR CUT(k) to reduce instances in order to then allow a reduction to VERTEX COVER ABOVE MATCHING and

[2] $\mathcal{O}^*$-notation suppresses factors polynomial in the input size.

application of the randomized kernelization obtained in the present work. Further kernelization applications include a problem of connectivity preservation under edge deletion [37].

Organization. Section 2 contains general preliminaries, and Section 3 contains additional background material relating to representative sets. Section 4 presents the approach of polynomial kernelization via representative sets in more detail. Thereafter, Section 5 contains the results on cut-covering sets; Section 6 gives applications to the kernelization of ALMOST 2-SAT and related problems; and Section 7 gives applications to MULTIWAY CUT and GROUP FEEDBACK VERTEX SET problems. Section 8 finishes the paper.

2 Preliminaries

Parameterized complexity and kernelization A *parameterized problem* is a language $\mathcal{Q} \subseteq \Sigma^* \times \mathbb{N}$; the second component of instances (x, k) is called the parameter (cf. [21, 14]). A parameterized problem is *fixed-parameter tractable* (FPT) if there is an algorithm A and a computable function $f \colon \mathbb{N} \to \mathbb{N}$ such that A decides $(x, k) \in \mathcal{Q}$ in time $f(k)|x|^{\mathcal{O}(1)}$. A *kernelization of* $\mathcal{Q}$ is a polynomial-time computable mapping $K \colon \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N} \colon (x, k) \mapsto (x', k')$ such that $|x'|, k' \leq h(k)$ and such that $(x, k) \in \mathcal{Q}$ if and only if $(x', k') \in \mathcal{Q}$; here h is a computable function that is called the *size* of the kernel and K is a *polynomial kernelization* if $h(k)$ is polynomially bounded.

Occasionally, we will use as shorthand a phrase of *accepting* or *rejecting* an instance of a parameterized problem in a kernelization. Formally, this refers to producing an output (x', k') that is, respectively, a constant-sized positive or negative instance of the problem.

A *compression* of a parameterized problem Q, into a (not necessarily parameterized) problem Q', is a polynomial-time computable mapping $K \colon \Sigma^* \times \mathbb{N} \to \Sigma^* \colon (x, k) \mapsto x'$ such that $|x'| \leq h(k)$ and such that $(x, k) \in \mathcal{Q}$ if and only if $x' \in \mathcal{Q}$. Again, h is a computable function that is called the *size* of the compression and K is a *polynomial compression* if $h(k)$ is polynomially bounded. Note that a polynomial kernelization implies a polynomial compression. Conversely, if Q' is in NP, and if Q is NP-complete when k is coded in unary, then a polynomial compression implies a polynomial kernelization [9].

Fortnow and Santhanam [31] and Bodlaender et al. [6] provided a framework to exclude the existence of a polynomial kernelization for a parameterized problem under a standard complexity-theoretical assumption. More generally, the framework excludes polynomial compressions, as well as more general notions including non-uniform compressions; see Dell and van Melkebeek [19].

All kernelization results in this paper are randomized, i.e., there is a (small) chance for the reduced instance not to be equivalent to the input. In all cases, such failure occurs with probability exponentially small in the input size. Such kernelizations are easily seen to be computable in non-uniform polynomial time using Adleman's trick [2], hence they are compatible with the above-mentioned framework [31], i.e., a successful application of the lower bound framework would (conditionally) rule out kernelizations such as ours. The interested reader is also referred to the work of Drucker [22] regarding lower bounds for randomized compression.

Let $P, Q \subseteq \Sigma^* \times \mathbb{N}$ be parameterized problems. A *polynomial parameter transformation* (PPT) from P to Q is a polynomial-time computable mapping $K \colon \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N} \colon (x, k) \mapsto (x', k')$ such that $(x, k) \in P$ if and only if $(x', k') \in Q$, and $k' \leq p(k)$ for some polynomial p. In this case, a polynomial kernel for Q implies a polynomial compression for P, and under the same conditions as above additionally implies a polynomial kernel for P. [10]

Graphs In this work, graphs $G = (V, E)$ may be undirected, directed, or mixed, i.e., $E \subseteq \binom{V}{2}$, $E \subseteq V^2$, or $E \subseteq \binom{V}{2} \cup V^2$. Some terms and statements are given for mixed graphs, as those strictly contain undirected and directed graphs as special cases; aspects specific to the purely undirected or directed case are pointed out where applicable. We reserve $D = (V, A)$, with $A \subseteq \{(u, v) \mid u, v \in V \wedge u \neq v\}$, to distinguish directed graphs in a few places.

A path from s to t in a mixed graph is a sequence $(s = v_0, v_1, \ldots, v_\ell = t)$ such that for each $i \in \{0, \ldots, \ell - 1\}$ we have $\{v_i, v_{i+1}\} \in E$ or $(v_i, v_{i+1}) \in E$; in other words, any undirected edge $\{u, v\}$ can be traversed in both directions, any directed edge (u,v) only from u to v. We are mainly interested in *vertex cuts* of such graphs and there are two prevalent forms: (1) For given sets $A, B \subseteq V$, an *(A,B)-cut* is a set $X \subseteq V$ such that there is no path from $A \setminus X$ to $B \setminus X$ in $G - X$; we note that paths from $B \setminus X$ to $A \setminus X$ may still exist in $G - X$. We also emphasize that, throughout the paper, (A,B)-cuts may intersect A and B, unless otherwise noted. As a shorthand, we write (v,B)-cut for the case where $A = \{v\}$ and the cut may *not intersect* A. Throughout we use (A,B)-min cut to refer to an (A,B)-cut of minimum cardinality. (2) For a given set $T \subseteq V$ and a partition $\mathcal{T} = \{T_1, \ldots, T_\ell\}$ of T, a $\mathcal{T}$-multiway cut is a set $X \subseteq V$ (optionally restricted to $X \subseteq V \setminus T$) such that there is no s,t-path in $G - X$ with s and t in different sets T_i and T_j; we note that this excludes paths from T_i to T_j and vice versa for all $1 \leq i < j \leq \ell$. In particular, the notions of (A,B)-cut and $\{A,B\}$-multiway cut are distinct.

Because we are working only with vertex cuts, the above definitions are fully consistent with simulating all graphs as purely directed graphs by replacing any undirected edge $\{u,v\}$ with its two directed counterparts (u,v) and (v,u) (unless already present). Thus, all our results for directed graphs hold just as well for undirected and mixed graphs. Our results for undirected graphs, however, especially regarding multiway cuts and related problems, do not extend to directed or mixed graphs: One crucial obstruction is that there is no polynomial kernelization (under standard assumptions) for DIRECTED 2-MULTIWAY CUT (k) [15], equivalently, for breaking all paths from A to B and simultaneously all paths from B to A in a directed graph using at most k vertex deletions.

Most of our techniques involve adding additional vertices with only incoming or only outgoing directed edges. Generally, this does not add or remove any paths (in the above sense) between vertices of the original graph, but it helps to formalize existence of certain path packings. If the given graph is undirected then this will turn it into a mixed graph but paths between original vertices still exist as before and, crucially, always in both directions.

We will heavily use the following immediate extension of Menger's Theorem [57] to mixed graphs. For ease of presentation we will usually switch freely between the size of an (A,B)-min cut and the maximum number of vertex-disjoint paths from A to B.

Proposition 1. *Let $G = (V,E)$ be a mixed graph and let $A, B \subseteq V$ be not necessarily disjoint vertex sets. The size of (A,B)-min cuts is equal to the maximum number of (fully) vertex-disjoint paths from A to B in G. Note that for $v \in A \cap B$ we have (v) as a legal path (of length zero).*

Matroids A *matroid* is a pair $M = (E, \mathcal{I})$, where E is the *ground set* and $\mathcal{I} \subseteq 2^E$ a collection of subsets of E, called *independent sets*, such that: (i) $\emptyset \in \mathcal{I}$; (ii) if $I_1 \subseteq I_2$ and $I_2 \in \mathcal{I}$, then $I_1 \in \mathcal{I}$; and (iii) if $I_1, I_2 \in \mathcal{I}$ and $|I_2| > |I_1|$, then there exists some $x \in (I_2 \setminus I_1)$ such that $I_1 \cup \{x\} \in \mathcal{I}$. A set $I \subseteq E$ is *independent* if $I \in \mathcal{I}$, and *dependent* otherwise. A set $B \in \mathcal{I}$ is a *basis* of M if no superset of B is independent; a matroid may equivalently be defined by its set of bases. For a subset $X \subseteq E$, the *rank* $r(X)$ of X is the largest cardinality of an independent set $I \subseteq X$. The rank of M is $r(M) := r(E)$.

Let A be a matrix over a field $\mathbb{F}$ and E be the set of columns of A. Let $\mathcal{I}$ be the set of all sets $X \subseteq E$ of columns that are linearly independent over $\mathbb{F}$ (as vectors). Then $(E, \mathcal{I})$ defines a matroid M, and we say that A *represents* M. A matroid is *representable* (over a field $\mathbb{F}$) if there is a matrix (over $\mathbb{F}$) that represents it. A matroid representable over some field is called *linear*. In this work, we will concern ourselves only with linear matroids. It is well known that row operations on a matrix, say A, do not affect the sets of columns of A that are linearly independent. Thus, any representation A for a matroid M of rank $r = r(M)$ can be reduced to one where A has only r rows: Pick a row basis of A and use row operations to reduce all other rows to all-zero; then discard all other rows without affecting the sets of independent columns (or, in fact, discard all but a row basis right away); see also Marx [53]. We will always assume that given representations are reduced in this sense.

For a ground set V and integer $r \in \mathbb{N}$, the *uniform matroid* of rank r is $U_{r,n} = (V, \binom{V}{r})$, i.e., the matroid where every set of size at most r is independent. The uniform matroid is representable over any sufficiently large field (indeed, it is a *gammoid*, but a representation can also be produced deterministically via a Vandermonde matrix, cf. [53]).

For two matroids $M_1 = (V_1, \mathcal{I}_1)$, $M_2 = (V_2, \mathcal{I}_2)$ on disjoint ground sets, the *direct sum* of M_1 and M_2 is the matroid $M_1 \oplus M_2 = (V_1 \cup V_2, \mathcal{I})$ where $I \in \mathcal{I}$ if and only if $I \cap V_i \in \mathcal{I}_i$, $i = 1, 2$. Naturally, the construction can be generalized to the direct sum over several matroids, with pairwise disjoint ground sets. Given representations A_1 of M_1 and A_2 of M_2 over the same field $\mathbb{F}$, a representation A of $M_1 \oplus M_2$ can be constructed as $A = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}$.

Gammoids Let $D = (V, A)$ be a directed graph and let $S, T \subseteq V$. The set T is *linked to* S if there exist $|T|$ vertex-disjoint paths from S to T; this allows paths of length zero, e.g., any set is linked to itself. Given any directed graph $D = (V, A)$ with source vertices $S \subseteq V$ and sink vertices $U \subseteq V$, the sets $T \subseteq U$ that are linked to S in D form a matroid, a so-called *gammoid* [65] (cf. [64]). We require only the special case of *strict gammoids* where $U = V$ but use the shorter name gammoid and use (D, S) to mean the (strict) gammoid on graph D with sources S (and sinks $U = V$). Marx [53] gave a randomized polynomial-time procedure for finding a representation of a strict gammoid. The error probability can be made exponentially small in the size of the graph.

Theorem 7 ([65, 53])**.** *Let $D = (V, A)$ be a directed graph, and let $S \subseteq V$. The subsets $T \subseteq V$ which are linked to S form the independent sets of a matroid over V. Furthermore, a representation of this matroid can be obtained in randomized polynomial time with one-sided error.*

To extend the notion of a gammoid to undirected and mixed graphs, we extend the notion of linked sets in the natural way: A set $T \subseteq V$ is *linked to* $S \subseteq V$ in an undirected or mixed graph $G = (V, E)$ if there are $|T|$ vertex-disjoint paths from S to T in G. Here, as before, paths may use undirected edges $\{u, v\}$ as either (u, v) or (v, u). Gammoids and strict gammoids for undirected and mixed graphs are now defined as in the directed case; in particular, we use (G, S) for the gammoid of G with source vertices S. This extension is equivalent to forming gammoids for $G = (V, E)$ using the directed graph $D = (V, A)$ where we have $(u, v), (v, u) \in A$ for all $\{u, v\} \in E$.

For an undirected, directed, or mixed graph $G = (V, E)$ and sets $S, T \subseteq V$, we write $\lambda_G(S, T)$ for the maximum number of vertex-disjoint paths from S to T in G, i.e., the unit-capacity vertex-disjoint flow from S to T. Equivalently, $\lambda_G(S, T)$ is the rank of T in the gammoid (G, S).

Note that for any sets $S, X \subseteq V$ in a graph $G = (V, E)$, the set X is linked to S if and only if $(X \setminus S)$ is linked to $(S \setminus X)$ in $G - (S \cap X)$. Hence in a gammoid (G, S), specifying members of S in an independent set X can be seen as "disabling" these vertices.

We use two crucial operations on graphs G to get a specific behavior in the gammoid (G, S):

1. Creation of *sink-only copies* of vertices. By adding a sink-only copy of a vertex v we mean adding a vertex v' that retains only the incoming edges of v (for undirected edges $\{u, v\}$ only the directed edge (u, v') is kept). These will be used, effectively, to require two paths to a vertex $v \in X$ in a set X linked to S. Note that adding v' to the graph has no effect on any independent (linked) set not containing v' because v' cannot be an internal vertex of any path. This will turn an undirected into a mixed graph, though the induced graph on the original vertex set remains the same.

2. By *bypassing a vertex v in a graph $G = (V, E)$* (or *making a vertex v undeletable in G*) we mean an operation that removes the vertex v from the graph while not changing the separation achieved by any vertex cuts that avoid v; this is achieved by adding suitable shortcut edges between the neighbors of v. Intuitively, for all paths (u, v, w) in G we ensure that (u, w) is a path in the resulting graph G' by adding the edge (u, w) or $\{u, w\}$ unless already present (the latter only if also (w, v, u) is a path in G). The exact implementation differs depending on the type of edges already in G so that, in particular, undirected and directed graphs are preserved: If $G = (V, E)$ only has directed edges then we add (u, w) to G' for each path (u, v, w); the result is a directed graph. If $G = (V, E)$ has only undirected edges then a path (u, v, w) in G implies that also (w, v, u) is a path in G; we add $\{u, w\}$ to G' for all such paths and obtain an undirected graph. For all other graphs, i.e., mixed graphs with both directed and undirected edges, we add (u, w) to G' for each path $(u, v, w) \in G$. (Note that we do not need the latter

case in our results.) We emphasize that from the perspective of vertex cuts and vertex-disjoint paths all variants of bypassing are functionally the same.

For an undirected graph $G = (V, E)$, bypassing all vertices in any set $S \subseteq V$ is equivalent to taking the so-called *torso* of G onto $V \setminus S$. Intuitively, this takes $G[V \setminus S$ and for all $u, v \in V \setminus S$ it adds the edge $\{u, v\}$ if there is a u, v-path in G whose internal vertices are all in S.

Intuitively, bypassing is only needed in the original directed or undirected graphs in our applications. There, we take the given graph, attach sink-only copies, construct a gammoid, and then identify a vertex v to bypass. We perform the latter in the original graph and start over.

Let us state formally the effects of bypassing a vertex.

Proposition 2. *Let $G = (V, E)$ be an undirected, directed, or mixed graph and let G' be the result of bypassing some vertex $v \in V$ in G. Then for any set $X \subseteq V \setminus \{v\}$ and any vertices $s, t \in V \setminus (X \cup \{v\})$, there is a path from s to t in $G - X$ if and only if there is a path from s to t in $G' - X$.*

Proof. Assume first that $G = (V, E)$ is an undirected graph, i.e., $E \subseteq \binom{V}{2}$, and let G' be obtained by bypassing v. Moreover, let $X \subseteq V \setminus \{v\}$ and let $s, t \in V \setminus X$. For the *if-part*, assume that $P = (s = v_0, v_1, \ldots, v_{\ell-1}, v_\ell = t)$ is an s, t-path in $G - X$. If P does not contain v then it is also an s, t-path in $G' - X$. Otherwise, let $i \in \{1, \ldots, \ell - 1\}$ so that $v = v_i$. It follows directly that $v_{i-1}, v_{i+1} \in N_G(v)$. Thus, G' contains the edge $\{v_{i-1}, v_{i+1}\}$ and, hence, $P' = (s = v_0, v_1, \ldots, v_{i-1}, v_{i+1}, v_{\ell-1}, v_\ell = t)$ is an s, t-path in $G' - X$. For the *only if-part*, assume that $P' = (s = v_0, v_1, \ldots, v_{\ell-1}, v_\ell = t)$ is an s, t-path in $G' - X$. If all edges of P', i.e., all edges $\{v_i, v_{i+1}\}$ with $i \in \{0, \ldots, \ell - 1\}$, are also present in G then P' is also an s, t-path in $G - X$. Otherwise, every edge $\{v_i, v_{i+1}\} \notin E$ can be replaced as follows: By definition of bypassing v, both v_i and v_{i+1} must neighbors of v in order for $\{v_i, v_{i+1}\}$ to be added in G'. Thus, we can replace each such edge by tracing P' through v using the edges $\{v_i, v\}$ and $\{v, v_{i+1}\}$ in $G - X$. If this happens more than once, then the resulting path, say $\hat{P}$, would contain multiple copies of v; clearly, $\hat{P}$ can be shortcut to an s, t-path P in $G - X$.

The argument is essentially the same for directed and mixed graphs, respectively: In the *if part*, if $v = v_i$ then (v_{i-1}, v, v_{i+1}) is a path in G and we have, thus, added $\{v_{i-1}, v_{i+1}\}$ or (v_{i-1}, v_{i+1}) to G'. In either case, the path $P' = (s = v_0, v_1, \ldots, v_{i-1}, v_{i+1}, v_{\ell-1}, v_\ell = t)$ is an s, t-path in $G' - X$. In the *only if-part*, we have (v_i, v_{i+1}) or $\{v_i, v_{i+1}\}$ in G' for all $i \in \{0, \ldots, \ell - 1\}$. If neither $(v_i, v_{i+1}) \in E$ nor $\{v_i, v_{i+1}\} \in E$ then (v_i, v, v_{i+1}) is a path in G, as that is required for either edge to be added to G'. But then we can replace the edge in question by tracing the path through v in $G - X$, and we can shortcut the resulting path $\hat{P}$ if necessary to get an s, t-path in $G - X$. $\square$

Observe that this proposition has a strong implication for vertex cuts X in G: If X does not contain v then X separates the same vertex pairs s, t (except vertex v) in G and in G'. In particular, achieving any separation request (e.g., a multiway cut for a set $T \subseteq V$) cannot have a cheaper solution after bypassing any vertex v; it could, however, become more expensive (or even impossible) if *all* minimum cuts contain vertex v. We will use the representative sets lemma and analysis of properties of vertex cuts to later find vertices v that can be bypassed while preserving at least one minimum cut for the given separation request, i.e., we will guarantee that there is at least one such cut that does not contain v, ensuring that the minimum cost to separate does not change.

3 Tools from matroid theory

3.1 Representative sets

The notion of *representative sets* plays an essential role in the paper.

Definition 1 ([53])**.** Let $M = (V, \mathcal{I})$ be a matroid and let X be an independent set in M. We say that a set $Y \subseteq V$ *extends* X in M if $X \cap Y = \emptyset$ and $X \cup Y \in \mathcal{I}$. For $\mathcal{Y} \subseteq 2^V$, we say that a subset $\mathcal{Y}^* \subseteq \mathcal{Y}$ is *r-representative for $\mathcal{Y}$* if the following holds: for every set $X \subseteq V$ of size at most r, if there is a set $Y \in \mathcal{Y}$ that extends X in M, then there is a set $Y' \in \mathcal{Y}^*$ that extends X in M.

We will use the term *representative set* without specifying r to mean an $(r(M)-s)$-representative set. The following result about representative sets is due to Marx [53], building on work of Lovász [51]; in this work, we refer to it as the *representative sets lemma*.

Lemma 2 (representative sets lemma [51, 53])**.** *Let M be a linear matroid of rank $r+s$, and let $\mathcal{Y}$ be a collection of independent sets of M, each of size s. There exists a set $\mathcal{Y}^* \subseteq \mathcal{Y}$ of size at most $\binom{r+s}{s}$ that is r-representative for $\mathcal{Y}$. Furthermore, given a representation A of M, we can find such a set $\mathcal{Y}^*$ in time $(m+||A||)^{\mathcal{O}(s)}$, where $m=|\mathcal{Y}|$ and $||A||$ denotes the encoding size of A.*

The running time was improved by Fomin et al. [28] to $(m+||A||)^{\mathcal{O}(1)}$. However, in all our applications s will be constant, in which case the above algorithm runs in polynomial time.

The following lemma is simple but useful.

Lemma 3. *Let $M=(V,\mathcal{I})$ be a matroid. For $i=1,\ldots,s$, let $M(i)$ denote a copy of M on a disjoint copy $V(i)$ of the ground set V, and let $M' = M(1)\oplus\ldots\oplus M(s)$ be their direct sum. Let $Y\subseteq V$ with $|Y|=s$, and write $Y=\{v_1,\ldots,v_s\}$. Then for any $X\subseteq V$, $\{v_i\}$ extends X in M for every $i\in[s]$ if and only if $\{v_1(1),\ldots,v_s(s)\}$ extends $X(1)\cup\ldots\cup X(s)$ in M', where $v(i)$ for $v\in V$ denotes the copy of v in $V(i)$, and $X(i)=\{v(i)\mid v\in X\}$.*

A similar statement applies if Y is a constant-sized tuple of constant-sized subsets of V (e.g., if we ask that X is extended by S_i for every $S_i\in Y$). For this situation, we can achieve an upper bound on the size of a representative set that is stronger than for the general case as given by the representative sets lemma; it follows by observing a nontrivial, smaller upper bound for the corresponding exterior space.

Lemma 4. *Let $M_1=(V_1,\mathcal{I}_1),\ldots,M_d=(V_d,\mathcal{I}_d)$ be a sequence of matroids, represented over the same field $\mathbb{F}$ and with pairwise disjoint ground sets, and let $V=V_1\cup\ldots\cup V_d$. Let $\mathcal{Y}\subseteq 2^V$ be a collection of sets $Y\subseteq V$ such that $|Y\cap V_i|=c_i$ for each $i\in[d]$ and each $Y\in\mathcal{Y}$, for some sequence $c_1,\ldots,c_d\in\mathbb{N}$. Then a representative set of $\mathcal{Y}$ in $M=M_1\oplus\ldots\oplus M_d$ has size at most $\prod_{i=1}^d\binom{r(M_i)}{c_i}$, where $r(M_i)=: r_i$ is the rank of M_i.*

Proof. Let $r=r_1+\ldots+r_d$ be the rank of M, and let $c=c_1+\ldots+c_d$ be the cardinality of sets $Y\in\mathcal{Y}$. Assume w.l.o.g. a representation of M where each $v\in V$ is represented by a vector in $\mathbb{F}^r$. Following the algorithm behind the representative sets lemma, as presented by Marx [53], we associate with every $Y\in\mathcal{Y}$ a c-vector $\bigwedge_{v\in Y} v$ in the exterior algebra over $\mathbb{F}^r$. The size of a representative set then corresponds to the dimension of the space spanned by these c-vectors, which is therefore the quantity we wish to bound.

For this, let $e_1,\ldots,e_r$ be a basis of M, consisting of a basis for M_i for every $i\in[d]$ in turn. We may then expand $\bigwedge_{v\in Y} v$ into a sum of wedge products $e_{i_1}\wedge\ldots\wedge e_{i_c}$ where each e_{i_j} is a base vector and $i_1<\ldots<i_c$. Here we use that the wedge product is antisymmetric, i.e.,

$$v_1\wedge\ldots\wedge v_{i-1}\wedge v_i\wedge v_{i+1}\wedge v_{i+2}\wedge\ldots\wedge v_r = -(v_1\wedge\ldots\wedge v_{i-1}\wedge v_{i+1}\wedge v_i\wedge v_{i+2}\wedge\ldots\wedge v_r),$$

and multi-linear, i.e.,

$$v_1\wedge\ldots\wedge v_{i-1}\wedge\left(\sum_{j=1}^{\ell}\alpha_j u_j\right)\wedge v_{i+1}\wedge\ldots\wedge v_r=\sum_{j=1}^{\ell}\alpha_j(v_1\wedge\ldots\wedge v_{i-1}\wedge u_j\wedge v_{i+1}\wedge\ldots\wedge v_r),$$

(cf. [73]). It is easy to observe that this produces only terms that for each $i\in[d]$ contain exactly c_i elements from the basis of M_i. The number of such terms is bounded by $\prod_{i=1}^d\binom{r_i}{c_i}$, which therefore is also a bound on the dimension and on the size of the resulting representative set. $\square$

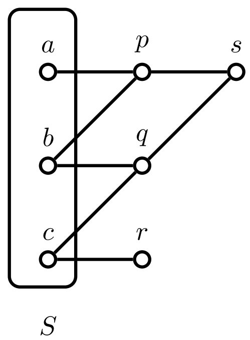


Figure 1: Illustration of closest sets. The sets $\{p, q\}$, $\{s, c\}$ and $\{a, b, c\}$ (among others) are closest sets to $S = \{a, b, c\}$. The set $\{p, q, r\}$ is not, because $\{a, b, c\}$ is a cut closer to S; the set $\{p, q, s\}$ is not, because $\{p, q\}$ is a smaller cut; and the set $\{p, s\}$ is not, because $\{p, q\}$ is a cut closer to S.

3.2 Closest cuts and gammoid rank

Our usage of representative sets and the representative sets lemma is centered around the concept of *closest sets*, defined as follows. See Figure 1 for an illustration. Let $G = (V, E)$ be a mixed graph and let $S \subseteq V$. A set $X \subseteq V$ is *closest to* S if X is the unique (S, X)-min cut in G. (Recall that an (S, X)-cut is permitted to intersect both S and X, and necessarily has to contain $S \cap X$.) If so, we say that X is a *closest set*. For any set of vertices X, the *induced closest set* $C(X)$ is the unique (S, X)-min cut that is closest to S. This is well-defined by the submodularity of cuts, cf. Schrijver [70, Chapter 44]. If X is a closest set, then $C(X) = X$.

More precisely, define an auxiliary graph G' by adding a new vertex s with $N^+(s) = S$ and $N^-(s) = \emptyset$, and define a function $f\colon 2^{V \setminus X} \to \mathbb{N}$ by $f(U) = |N^+_{G'}(U \cup \{s\})|$. Then f is submodular, i.e., it satisfies the inequality

$$f(A) + f(B) \geq f(A \cap B) + f(A \cup B)$$

for any $A, B \subseteq V \setminus X$. It follows that there is a unique minimal set U that minimizes f, and $N^+_{G'}(U \cup \{s\})$ is the closest min-cut $C(X)$. The induced closest set can be found efficiently by standard max-flow arguments. For this, introduce a further new vertex t with $N^-(t) = X$ and $N^+(t) = \emptyset$; replace each undirected edge $\{u, v\}$ by the two arcs (u, v), (v, u) to create a directed graph; split every vertex $v \in G' - \{s, t\}$ into two vertices v^-, v^+ with an arc (v^-, v^+), where arcs (u, v) into v are replaced by (u, v^-) and arcs (v, w) out of v are replaced by (v^+, w); and compute a max-flow from s to t in the resulting network, with arc capacities $c(v^-, v^+) = 1$ for all $v \in V$ and arc capacities 2 for all other arcs. Let G_f be the corresponding residual network, where arcs of capacity 0 have been deleted. Then $C(X)$ is precisely the set of vertices v such that v^- is reachable from s in G_f but v^+ is not.

Note that a closest set does not need to be a cut; there may not be any vertices except for X that are separated from S by X. In case of ambiguity, we also write $C_G(X, S)$ for $C(X)$ with the graph G and/or the set S specified.

Let us briefly clarify the special case of sets X that are closest to a single vertex x. By our convention, (v, B)-cuts may intersect B but not $A = \{v\}$. Accordingly, X is closest to v if X is the unique (v, X)-min cut *that does not contain* v. Note that this behavior can be simulated by creating sufficiently many parallel copies of v, namely more than $|X|$, and thus all further properties of closest sets go through for this special case as well.

Closest sets are a natural notion, but do not seem to have a fixed name; e.g., they occur in the bipedal stage of the MULTICUT(k) algorithm of Marx [54]. There are also similarities between closest sets and the concept of *important separators* [52]: for any set X closest to S that separates some vertex $v \notin X$ from S, there is a corresponding important separator. However, the change of focus from separation to closeness means that the concepts behave differently.

We make the following observations connecting closest sets to the rank function of a gammoid.

Proposition 3. *Let $G = (V, E)$ be a mixed graph with a set of sources $S \subseteq V$, and let $X \subseteq V$. Let G' be the result of adding a sink-only copy x' for every vertex $x \in X$. The following hold.*

1. *The set X is closest to S in G if and only if $X + x'$ is independent in the gammoid (G', S) for every $x \in X \setminus S$.*
2. *Let X_B be a maximal independent subset of X containing $X \cap S$. A vertex $v \in V$ is reachable from $S \setminus X$ in $G - C(X)$ if and only if $X_B + v$ is independent in the gammoid (G', S).*
3. *In particular, if X is independent and disjoint from S then any $v \in V$ is reachable from S in $G - C(X)$ if and only if $X + v$ is independent in (G', S).*

Proof. For part (1), assume first that the *only if-part* holds. Furthermore, assume for contradiction that X is not closest to S in G and let $Z \neq X$ be an (S, X)-min cut of size at most $|X|$ in G. Since $|Z| \leq |X|$ and $Z \neq X$, we can pick a vertex $v \in X \setminus Z$. Because Z is an (S, X)-cut it holds that $x \notin S$ and, hence, that $X + x'$ is independent in the gammoid. This implies that there are $|X| + 1$ paths from S to X that are vertex-disjoint except for sharing x. Thus, because $x \notin Z$, the set Z must miss at least one of the paths; a contradiction. Thus Z cannot exist and X is indeed closest to S in G.

For the *if-part* of (1), assume that X is closest to S in G. Thus, there is no (S, X)-cut of size less than $|X|$ in G and, by Proposition 1 (Menger's Theorem for mixed gaphs), there are $|X|$ vertex-disjoint paths from S to X in G. The same paths exist also in G' and, thus, X is independent in the gammoid (G', S). Now, let $x \in X \setminus S$ and assume for contradiction that $X + x'$ is dependent in the gammoid, implying that there are exactly $|X|$ vertex-disjoint paths from S to $X + x'$ in G' (as we already have $|X|$ paths from S to X in G'). Again by Menger's Theorem for mixed graphs, there is an $(S, X + x')$-cut Z of size $|X|$ in G'. We show that $x \notin Z$: The set Z must contain exactly one vertex from each of the $|X|$ vertex-disjoint paths from S to X in G', and no further vertices. Since x' is sink-only, it cannot be an internal vertex of such a path and, thus, it cannot be in Z. Now if $x \in Z$ then by minimality of Z there is a path from S to x that avoids $Z - x$. However, this path can also be used as a path from S to x' avoiding Z, contradicting that Z is an $(S, X + x')$-cut; thus $x \notin Z$. Clearly, Z is also an (S, X)-cut in G. Thus, Z is an (S, X)-cut of size $|X|$ but different from X (as $x \notin Z$), contradicting that X is closest to S in G. Thus, $X + x'$ is independent in the gammoid for all $x \in X \setminus S$.

For the *only if-part* of (2), assume that $X_B + v$ is independent. Thus, there is a set $\mathcal{P}$ of $|X_B| + 1$ vertex-disjoint paths from S to $X_B + v$ in G', including a path P_v from S to v. Since sink-only copies can only be endpoints of paths, all these paths exist also in G, including the set $\hat{\mathcal{P}} = \mathcal{P} \setminus \{P_v\}$ of $|X_B|$ vertex-disjoint paths from S to X_B. As, $C(X)$ separates S from $X \supseteq X_B$ it must intersect all $|X_B|$ vertex-disjoint paths in $\hat{\mathcal{P}}$. Thus, it has size exactly $|X_B|$ and contains exactly one vertex per path from S to X_B in $\hat{\mathcal{P}}$; it contains no further vertices. In particular, it does not contain any vertex on the path P_v from S to v, implying that v is reachable from $S \setminus X$ (as $S \cap X \subseteq X$) in $G - C(X)$.

For the *if-part* of (2), assume that $X_B + v$ is dependent. Since $|X_B| = r(X_B) = rank(X)$ and $|X_B| = r(X_B) = r(X_B + v)$, it follows that $r(X + v) = |X_B|$. Hence, the maximum number of vertex-disjoint paths from S to $X + v$ in G' is $|X_B|$, which implies by Menger's Theorem that there is an $(S, X + v)$-cut Z in G' of size $|X_B|$. Clearly, the set Z is also an $(S, X + v)$-cut in G (and also an (S, X)-cut). Now, $C(X)$ is either equal to Z or it is an (S, Z)-cut; either way, it separates S from v.

The third part follows as a special case of the second. □

A further simplified version of the above is often useful: Let $G = (V, E)$ be a mixed graph and let $S \subseteq V$ be a set of source vertices. Let $X \subseteq V$ be closest to S in G. Then for every $x \in V$, the set $\{x\}$ extends X in the gammoid (G, S) if and only if x is reachable from S in $G - X$.

Also note that if an (S, T)-cut X is not a closest set, i.e., if $X + x'$ is dependent in the gammoid (G', S) for some $x \in X \setminus S$, then X can be replaced by a different (S, T)-cut Z of at most the same size which does not contain x, namely by $Z = C(X, S)$: Indeed, any path P from S to T must pass through X but then the part P' from S to X also contains a vertex of Z. Thus, Z is indeed an (S, T)-cut.

4 Kernelization usage of representative sets

In this section, we show different ways of using the representative sets lemma for kernelization purposes. We divide the usage into three types – direct usage, indirect usage, and irrelevant vertex applications – and give a brief explanation and example for each part. Along the way, we illustrate using polynomial kernels for the d-HITTING SET and MULTIWAY CUT WITH DELETABLE TERMINALS problems (defined below). More involved applications of the framework will then follow in subsequent sections. See also the corresponding chapter of Cygan et al. [14] for a gentle presentation.

4.1 Direct usage

The most immediate kernelization application of the representative sets lemma that we will use in this paper is to reduce *constraint systems*. We illustrate this with an application to the d-HITTING SET problem. In d-HITTING SET, the input is a d-uniform hypergraph $\mathcal{H}$ with vertex set V (i.e., a collection of d-ary subsets of V) and an integer k, and the task is to find a set $X \subseteq V$ with $|X| \leq k$ such that $E \cap X \neq \emptyset$ for every $E \in \mathcal{H}$ (a *hitting set* of $\mathcal{H}$). This is a classical problem in parameterized complexity, and is known to have a polynomial kernel of $\mathcal{O}(k^d)$ edges and $\mathcal{O}(k^{d-1})$ vertices [27, 1], and no kernel with $\mathcal{O}(k^{d-\varepsilon})$ edges for any $\varepsilon > 0$ unless the polynomial hierarchy collapses [19].

The most classical polynomial kernel for d-HITTING SET uses the Erdős-Rado *sunflower lemma* [23, 27]. Using this lemma, one can in polynomial time reduce a given d-uniform hypergraph $\mathcal{H}$ with vertex set V to a d-uniform hypergraph $\mathcal{H}' \subseteq \mathcal{H}$ with $|\mathcal{H}'| \leq (k+1)^d \cdot d!$, such that for every $X \subseteq V$ with $|X| \leq k$, X is a hitting set of $\mathcal{H}$ if and only if it is a hitting set of $\mathcal{H}'$. (This variant was stated in [30]; the variant given in [27] produces an output hypergraph of slightly smaller size, containing sets of size *at most* d rather than exactly d.) We show how to use representative sets for the same conclusion, with improved size bound. The fact that this kernel gives an improved constant factor over [27] was first observed in the textbook of Cygan et al. [14, Section 12.3.2].

Theorem 8. *Let $\mathcal{H}$ be a d-uniform hypergraph with vertex set V, where d is a constant. In polynomial time, we can compute a hypergraph $\mathcal{H}' \subseteq \mathcal{H}$ such that for any $X \subseteq V$ with $|X| \leq k$, X is a hitting set of $\mathcal{H}$ if and only if X is a hitting set of $\mathcal{H}'$, and such that $|\mathcal{H}'| \leq \binom{k+d}{d} = (k+d)\ldots(k+1)/d! = k^d/d! + \mathcal{O}(k^{d-1})$. In particular, d-HITTING SET has a polynomial kernel with at most $\binom{k+d}{d} = k^d/d! + \mathcal{O}(k^{d-1})$ edges.*

Proof. Let $(\mathcal{H}, k)$ be the input to d-HITTING SET, and let V be the vertex set of $\mathcal{H}$. Let $M = (V, \binom{V}{k+d})$ be the uniform matroid over V with rank $k+d$; a representation by a $(k+d) \times |V|$ Vandermonde matrix can be constructed in polynomial time (cf. [53]). We claim that for any set $X \subseteq V$ with $|X| \leq k$, X is a hitting set of $\mathcal{H}$ if and only if there is no set $E \in \mathcal{H}$ such that E extends X. Indeed, on the one hand, if $E \in \mathcal{H}$ and E extends X, then (by definition) $E \cap X = \emptyset$, hence X is not a hitting set. On the other hand, if X is not a hitting set, then by definition there is some set $E \in \mathcal{H}$ such that $E \cap X = \emptyset$, in which case E extends X in M (in particular, $E \cup X$ is independent in M since $|E \cup X| \leq d + k$). Thus X is a hitting set of $\mathcal{H}$ if and only if it cannot be extended by any set $E \in \mathcal{H}$ in the matroid M.

Hence we do as follows. Let A represent M over some field $\mathbb{F}$. Apply the representative sets lemma to M and $\mathcal{H}$, and let $\mathcal{H}' \subseteq \mathcal{H}$ be the representative set. Then for any X as above, X can be extended by some $E \in \mathcal{H}$ if and only if it can be extended by some $E' \in \mathcal{H}'$, i.e., $\mathcal{H}'$ has the properties we need, and the bound on $|\mathcal{H}'|$ follows from the representative sets lemma. All operations can be performed in polynomial time. □

To further restrict the kernel to $\mathcal{O}(k^{d-1})$ vertices, we may simply apply the kernelization of Abu-Khzam [1] to the output. However, this does not maintain the guarantee of preserving *all* solutions $X \in \binom{V}{k}$.

The intended interpretation of the above (in particular the term *constraint system*) is that the hypergraph $\mathcal{H}$ over V represents a system of *constraints*, where every set $E \in \mathcal{H}$ represents the constraint "the solution X must intersect E," and a solution (a hitting set) is exactly a set X which satisfies all constraints. The set $\mathcal{H}' \subseteq \mathcal{H}$ then represents a constraint system equivalent to $\mathcal{H}$ for solutions up to cardinality k, as every constraint $E \in \mathcal{H} \setminus \mathcal{H}'$ is a consequence of constraints in $\mathcal{H}'$ together with the cardinality bound $|X| \leq k$.

A further application along these lines will be given in Section 6.1 for a problem we refer to as DIGRAPH PAIR CUT, occurring as a part of the kernelization of ALMOST 2-SAT. In this problem, the "constraints" are given in terms of reachability conditions in a directed graph, and the constraints reduction is achieved using representative sets over gammoids. In this case, some further complications arise due to the need to phrase the conditions in terms of *closest sets* (cf. Proposition 3); see Section 6.1 for more details. Unlike the d-HITTING SET example, the constraints reduction for DIGRAPH PAIR CUT is a result for which no purely combinatorial proof is known.

4.2 Indirect usage

For more advanced applications, we can "force" the representative sets lemma to reveal some set Z of special vertices in a graph G (e.g., a small superset of the set of vertices included in an optimal solution), by showing the existence of "queries" X that only vertices in Z can "answer". This can work as follows. Let M be a linearly represented matroid, and let $\mathcal{Y} = \{Y(v) \mid v \in V\}$ be a collection of subsets of M of cardinality s. Assume that we have shown that for every $z \in Z$ there is a carefully chosen set $X(z)$, such that $Y(v)$ extends $X(z)$ if and only if $v = z$. Then, necessarily, any representative set $\mathcal{Y}^*$ for $\mathcal{Y}$ must contain $Y(z)$ for every $z \in Z$, by letting $X = X(z)$ in Definition 1 (thus, $X(z)$ is the "query set" referred to above). Furthermore, we do not need to provide the set $X(z)$ ahead of time, since the (possibly non-constructive) *existence* of a set $X(z)$ that can only be extended by $Y(z)$ is sufficient to force $Y(z) \in \mathcal{Y}^*$. Hence the set $V^* = \{v \in V \mid Y(v) \in \mathcal{Y}^*\}$ must contain all vertices Z, among a polynomially bounded number of other vertices.

Note that the requirement here is twofold. We need to design a linear matroid M and collection of sets $Y(v)$, as well as a proof of existence for sets $X(z)$, such that (i) $Y(z)$ extends $X(z)$, and (ii) $Y(v)$ fails to extend $X(z)$ for every $v \neq z$. (Additionally, for a useful size bound we need that $|X(z)| = k^{\mathcal{O}(1)}$ and $|Y(v)| = \mathcal{O}(1)$.)

We illustrate the approach with a variant of MULTIWAY CUT referred to as MULTIWAY CUT WITH DELETABLE TERMINALS (DT-MWC(k)), where the task is to pairwise separate a set T of terminals in an undirected graph by deleting at most k vertices (including terminals). It can be easily seen to be equivalent to MULTIWAY CUT restricted to terminals of degree one. The problem is NP-hard by a simple reduction from VERTEX COVER: Given a graph G, create G' by attaching a terminal v' to each vertex v. Multiway cuts in G' correspond to vertex covers of G and vice versa.

Let (G, T, k) be an instance of DT-MWC(k). We kernelize the instance in two steps. First, we use known results to reduce T so that $|T| \leq 2k$; this step follows the LP-based approach of Guillemot [34] (also used in [17]) for reducing the number of terminal neighbors in instances of MWC(k), using the equivalence of DT-MWC(k) with MWC(k) restricted to terminals of degree one. Second, we use the above strategy to force the identification of a set V^* of $\mathcal{O}(k^3)$ vertices such that $V^* \cup T$ contains an optimal solution. We begin with the first step, showing how to adapt the result of Guillemot [34] to DT-MWC(k).

Lemma 5. *Let (G, T, k) be an instance of* DT-MWC(k). *An equivalent instance (G', T', k') with $k' \leq k$ and $|T'| \leq 2k'$ can be computed in polynomial time.*

Proof. Create an instance (G', T', k) of MULTIWAY CUT (MWC(k), with undeletable terminals) by adding a pendant vertex t' for every $t \in T$, and let T' be the set of all pendants added this way. Note that a set $X \subseteq V(G')$ is a valid multiway cut in G' if and only if $X \subseteq V(G)$ and X is a multiway cut in G; i.e., the two problems have identical solution sets.

Guillemot [34, Section 2.2] showed a procedure that computes a half-integral optimal solution to an LP-relaxation for MWC(k) that additionally has certain *persistence* properties. Let $\phi\colon V(G) \to \{0, 1/2, 1\}$ be the half-integral LP-optimum computed this way, and let $X_1 = \phi^{-1}(1)$ and $X_{1/2} = \phi^{-1}(1/2)$. Then every path between distinct terminals $t_i, t_j \in T'$ intersects X with weight at least 1, i.e., either at least one vertex of X_1 or at least two vertices of $X_{1/2}$. Furthermore, let $U \subseteq V(G)$ be the set of vertices reachable from T' in $G' - X$. Then Guillemot shows that there exists an optimal multiway cut X' of G' such that $U \cap X' = \emptyset$ and $X_1 \subseteq X'$.

Our reduction rule for DT-MWC(k) is now as follows. Compute the approximate solution $X = X_1 \cup X_{1/2}$ and the set U as above. If $|X_1| + \frac{1}{2}|X_{1/2}| > k$, reject the instance, since the LP-relaxation costs more than k. Otherwise, for every vertex $x \in X_1$, delete x from G and decrease k by 1, and for every vertex $u \in U \setminus T$, bypass u in G. Finally, discard any resulting connected components containing at most one terminal. Let (G'', T'', k') be the resulting instance. We claim that $|T''| \leq |X_{1/2}| \leq 2k'$. Indeed, note first that after the bypassing step, for every terminal $t \in T'' \setminus X_{1/2}$ we have $N_{G''}(t) \subseteq X_{1/2}$. Now consider a vertex $x \in X_{1/2}$. If x neighbours two distinct vertices of $T'' \setminus X_{1/2}$, or if $x \in T''$ and neighbours at least one such vertex, then G contains a terminal-terminal-path of weight $1/2$, which contradicts ϕ being a solution. It follows that $|T''| \leq |X_{1/2}|$, and since $|X_{1/2}| \leq 2(k - |X_1|) = 2k'$ we are done. □

Towards identifying (a superset of) an optimal solution we use a well-known fact about multiway cuts: For every terminal $t \in T$ there is an optimal multiway cut X that is closest to the set $T - t$ (respectively, closest to $N(T - t)$ when terminals are not deletable); e.g., this was used by Marx in an FPT algorithm for the problem [52]. Via Proposition 3, this entails a high reachability of X from $T - t$. In the case of DT-MWC(k) we are fortunate in that a simple criterion forces solutions to be closest to $T - t$ for all $t \in T \setminus X$, as shown by the following lemma.

Lemma 6. *Let (G, T, k) be an instance of* DT-MWC(k) *and let X be a minimum size multiway cut which as a secondary criterion has a maximum size intersection with T. Then, for every $t \in T \setminus X$, the set X is closest to $T - t$.*

Proof. Assume that X is not closest to $T - t$ for some $t \in T \setminus X$ and, thus, let C be a minimum cut separating X from $T - t$ with $C \neq X$ and $|C| \leq |X|$. Clearly, C is also a multiway cut for T in G; thus, $|C| = |X|$ as X is of minimum size. It follows that C contains at least one vertex $v \notin X$. By minimality of C there must be a path P from some $t_1 \in T - t$ to X that intersects C only in v. We can clearly assume that P intersects X only in its final vertex. Since $v \notin X$, we conclude that v is reachable from t_1 in $G - X$. Then in particular, since X is a multiway cut, no other terminal $t_2 \in T - t$ can reach v in $G - X$. It follows that $C' := (C \setminus \{v\}) \cup \{t_1\}$ also separates $T - t$ from X, i.e., C' is a multiway cut. We claim that $|C' \cap T| > |X \cap T|$, contradicting our choice of X. Indeed, since C' is an $(X, T - t)$-cut we have $X \cap (T - t) = X \cap T \subset C'$, while additionally $t_1 \in C' \setminus X$. Thus the cut C cannot exist, and X must be closest to $T - t$. □

Recall that we wish to identify (a polynomial-sized superset of) an optimal solution X. Since $|T| \leq 2k$ it suffices to identify the set $Z = X \setminus T$. To this end, the following lemma proves that all vertices in Z are highly reachable from T. This is expressed via path packings in the graph augmented with two sink-only copies v', v'' for each non-terminal vertex v.

Lemma 7. *Let (G, T, k) be an instance of* DT-MWC(k) *and let X be a minimum size multiway cut which as a secondary criterion has a maximum size intersection with T. Let G' be the result of adding two sink-only copies v', v'' for each non-terminal vertex v of G. Then, for every $x \in X \setminus T$, the set $X \cup \{x', x''\}$ is linked to T in G'.*

Proof. Fix any $x \in X \setminus T$. Since X is minimal there must be two terminals $t_1, t_2 \in T$ that have a t_1, t_2-path in $G - (X - x)$. It follows that there is a t_1, x-path whose vertices other than x are contained in the component of t_1 in $G - X$. By replacing the endpoint x with the sink-only copy x'' we get a t_1, x''-path P. Note that P is contained in G except for its endpoint.

Now, we use that X is closest to $T - t_1$ (by Lemma 6) which, by Proposition 3, implies that for every $x \in X \setminus T$ the set $X \cup \{x'\}$ is linked to $T - t_1$. Fix an appropriate path packing $\mathcal{P}$ of $|X| + 1$ paths from $T - t_1$ to $X \cup \{x'\}$ for analysis. Observe that every vertex of X is an endpoint of a path in $\mathcal{P}$. Thus, no path in $\mathcal{P}$ can have vertices of the component of t_1 in $G - X$ as internal vertices since that would require entering the component through X (and sink-only copies of vertices can obviously not be used for this). Hence, the t_1, x''-path P and the path packing $\mathcal{P}$ from $T - t_1$ to $X \cup \{x'\}$ are vertex-disjoint. It follows that $X \cup \{x', x''\}$ is linked to T, as claimed. □

Now we are set to apply the representative sets lemma. Let M be the gammoid defined by the graph G' as in the lemma and with source set T, and form the set $\mathcal{Y}$ of triples $Y(v) := \{v, v', v''\}$ for $v \in V \setminus T$. We will show that for a solution X as above, for every $x \in X \setminus T$, a triple $Y(v)$ extends $X - x$ in M if and only if $v = x$. We can then complete the kernel by computing a representative set for $\mathcal{Y}$ in M.

Let us fill in the details. Recall that the representative sets lemma takes a collection of s-tuples and a matroid of rank $r + s$, and computes a representative set of size $\binom{r+s}{s}$. By the above outline, we will need a $(k-1)$-representative set of $\mathcal{Y}$, i.e., we aim for an application with $r = k - 1$ and $s = 3$. Consider first that $r(M) < r + s = k + 2$. Then $|T| \leq k + 1$, and we can construct a solution $T - t$ of size k to solve the input in polynomial time. Otherwise, $|T| \geq k + 2$ and we have two options. We can either compute the *truncation* M' of M to rank $k + 2$ (see, e.g., Marx [52]), and apply the representative sets lemma to $\mathcal{Y}$ in M'; or we can observe that a representative set for $\mathcal{Y}$ in M will contain a $(k-1)$-representative set, and have size $\mathcal{O}(|T|^3) = \mathcal{O}(k^3)$ just like the intended one would.

In either case, let $\mathcal{Y}^* \subseteq \mathcal{Y}$ be the computed representative set, and let $V^* \subseteq V$ be the set of vertices v with $Y(v) \in \mathcal{Y}^*$. We know that $|V^*| = \mathcal{O}(k^3)$; we argue (again) that there is an optimal solution X such that $(X \setminus T) \subseteq V^*$.

Lemma 8. *Let (G, T, k) be an instance of* DT-MWC(k) *and let X be a minimum size multiway cut which as a secondary criterion has a maximum size intersection with T. Then $X \subseteq T \cup V^*$.*

Proof. Clearly, it suffices to prove that $x \in V^*$ for all $x \in X \setminus T$. Fix such a vertex x and recall, from Lemma 7, that $X \cup \{x', x''\}$ is linked to T. Thus, $Y(x) = \{x, x', x''\}$ extends $X - x$ (to an independent set in the constructed gammoid). We show that $Y(x) \in \mathcal{Y}^*$ by proving that among sets in $\mathcal{Y}$ it *uniquely* extends the set $X - x$.

Consider any vertex $v \neq x$ and assume that $\{v, v', v''\}$ extends $X - x$; this entails that $v \notin X$. A direct (but weaker) conclusion is that there must be three vertex-disjoint paths from T to $\{v, v', v''\}$ in $G - (X - x)$ since clearly any path packing from T to $(X - x) \cup \{v, v', v''\}$ has $X - x$ as endpoints of paths and cannot reuse them as internal vertices of paths to $\{v, v', v''\}$. But in $G - X$, v must be part of some connected component containing at most one terminal, say t. It follows that deletion of x and t in $G - (X - x)$ would separate v from *all terminals* (or possibly delete v itself if $v = t$). This is a contradiction to $\{v, v', v''\}$ being linked to T in $G - (X - x)$.

We conclude that $Y(x)$ uniquely extends $X - x$ among triples in $\mathcal{Y}$, implying that $\{x, x', x''\} \in \mathcal{Y}^*$ and $x \in V^*$. Thus, $X \setminus T \subseteq V^*$ and $X \subseteq T \cup V^*$, as claimed. ☐

By this lemma, we may set all vertices of $V' := V \setminus (V^* \cup T)$ as undeletable, without changing the existence of a solution of size at most k. These vertices can then be removed from the graph, using the *bypassing* (or *torso*) operation described in Section 2.

Theorem 9. DT-MWC(k) *has a randomized kernel of $\mathcal{O}(k^3)$ vertices. The error probability can be made exponentially small in n; all errors are false negatives.*

4.3 Irrelevant vertex strategies

For more involved applications of the vertex identification form of the representative sets lemma, we turn to *irrelevant vertex* strategies. The idea is as follows. Let $\mathcal{I}$ be an instance of a cut problem on a graph G, e.g., an instance of s-MULTIWAY CUT. Following the strategy of the previous subsection, we would seek to define query sets $X(z)$ and tuples $Y(v)$ in some linear matroid M, such that for every vertex z in an optimal solution, $Y(v)$ extends $X(z)$ if and only if $v = z$. However, there may be several incomparable optimal solutions, and it may not be feasible to pin down a list of criteria that perfectly captures any one solution. Instead, the goal is to identify the set of vertices contained in *every* optimal solution (referred to as *essential* vertices), and then select a single vertex v that is non-essential (i.e., *irrelevant*) and remove it from the graph. This is the basis of our kernel for s-MULTIWAY CUT; see Sections 5.2 and 7.

Here, we will illustrate using a simpler application of the process. Let $G = (V, E)$ be an undirected graph, and let $S, T \subseteq V$ be two disjoint sets of vertices. Assume $|S| = |T| = k$. We will seek to produce a specific kind of *mimicking network* for G, namely a graph $G' = (V', E')$ on $\mathcal{O}(k^3)$ vertices, where $S, T \subseteq V'$, such that for every pair of sets $(A \subseteq S, B \subseteq T)$, the size of a minimum (A,B)-vertex cut is identical in G and in G'. (This particular notion of mimicking network has been chosen in order to make the proof illustration as streamlined as possible. In Section 5 we use similar arguments in a more flexible manner to produce more standard mimicking networks, as well as the seemingly more powerful conclusion of cut-covering sets.)

We use the irrelevant vertex strategy outlined above, i.e., we repeatedly find a single vertex v that we can bypass without changing the minimum size of an (A,B)-cut for any (A,B). This is done by identifying a set Z of $\mathcal{O}(k^3)$ vertices that is guaranteed to contain every essential vertex, then picking some $v \in V \setminus (S \cup T \cup Z)$ as irrelevant. Once we cannot find such a vertex v, we thus have $|V'| = \mathcal{O}(k^3)$. The set Z, in turn, is identified using the indirect approach illustrated in Section 4.2.

Let us consider the notions of essential and irrelevant vertices a bit closer in this setting. Recall that $\lambda_G(A,B)$ for $A, B \subseteq V$ denotes the cardinality of a minimum (A,B)-vertex cut in G. Let $v \in V \setminus (S \cup T)$, and let G' be the result of bypassing v in G. By definition, we will consider v *essential* if there is some $A \subseteq S$ and $B \subseteq T$ such that $\lambda_G(A,B) \neq \lambda_{G'}(A,B)$. In this case, we say that v is *essential for (A,B)-cuts in G*.

Now note the following. By Proposition 2, for every (A,B)-vertex cut X in G', X is an (A,B)-vertex cut in G; conversely, if X is a minimal (A,B)-vertex cut in G, then X is an (A,B)-vertex cut in G' if and only if $v \notin X$. Hence, for every A and B we have $\lambda_G(A,B) \leq \lambda_{G'}(A,B)$, and the inequality is strict if and only if v is contained in *every* minimum (A,B)-vertex cut. We find that v is essential in G if and only if there are some sets $A \subseteq S$ and $B \subseteq T$ such that every minimum (A,B)-vertex cut in G contains v. Hence, following the strategy of Section 4.2, we wish to define a linear matroid M and collection of s-tuples $\mathcal{Y} = \{Y(v) : v \in V \setminus (S \cup T)\}$ such that for every pair of sets $(A \subseteq S, B \subseteq T)$ and every $v \in V$ contained in all minimum (A,B)-vertex cuts there exists an independent set $X(A,B,v)$ in M such that $Y(u)$ extends $X(A,B,v)$ if and only if $u = v$. We will gradually work our way towards such a result. We will omit some simple proofs; see Section 5 for a more formal presentation (of a stronger result). In particular, the general result applies to directed graphs. The restriction here to undirected graphs is merely for simplicity.

We begin with the following proposition. Recall from Section 2 that $C(X,S)$ denotes the *induced closest set* of X.

Proposition 4. *Let $G = (V,E)$ be an undirected graph, with vertex sets $A, B \subseteq V$. Let X be an (A,B)-vertex cut in G of minimum cardinality. Then for any vertex $v \in V \setminus (A \cup B)$, the following are equivalent.*

1. *v is essential for (A,B)-cuts in G;*
2. *v occurs in every minimum (A,B)-vertex cut in G;*
3. *$v \in C(X,A) \cap C(X,B)$.*

This can be translated into language suitable for the representative sets lemma as follows. Let $v \in C(X,A) \cap C(X,B)$ with $v \notin A \cup B$, and let v' be a sink-only copy of v. By Proposition 3(1), $C(X,A) + v'$ is linked to A, and $C(X,B) + v'$ is linked to B. On the other hand, it is not hard to see that vertices u' such that $u \in C(X,A) \cap C(X,B)$ are the only vertices with this property, as these are the only vertices that are reachable from both A and B avoiding $C(X,A)$ resp. $C(X,B)$. Hence $u \in C(X,A) \cap C(X,B)$ if and only if $\{u'\}$ extends both $C(X,A)$ in the gammoid (G,A) and $C(X,B)$ in the gammoid (G,B). Furthermore, there are at most k such vertices. Using Lemma 3, we can easily define an application of the representative sets lemma that will capture all such vertices u in a set Z (see below).

It remains to adapt the result to allow the free choice of sets $A \subseteq S$ and $B \subseteq T$. For this, create for every vertex $v \in S \cup T$ a pendant vertex v', i.e., a new vertex v' such that $N(v') = \{v\}$, where $\{v, v'\}$ is an undirected edge. Then for any $X \subseteq V$ and $A \subseteq S$, it holds that X is linked to A in G if and only if $X \cup \{v' : v \in (S \setminus A)\}$ is linked to $S' = \{v' : v \in S\}$ in the resulting graph (see Figure 2). Putting it all together we get the following result.

Proposition 5. *Let $G = (V,E)$ and $S, T \subseteq V$ be as above. Let $G' = (V', E')$ be the result of adding a sink-only copy v' for every $v \in V \setminus (S \cup T)$, and adding a pendant vertex v' for every $v \in S \cup T$. For a set $U \subseteq V$, let $U' = \{u' : u \in U\}$. For any pair $(A \subseteq S, B \subseteq T)$, any minimum (A,B)-vertex cut X, and any $v \in V \setminus (S \cup T)$, the following are equivalent.*

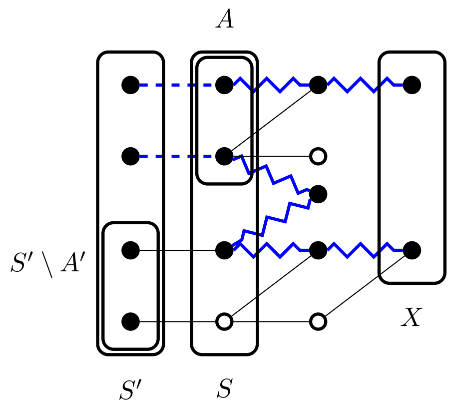


Figure 2: Illustration of Proposition 5. Paths linking X to $A \subseteq S$ (zig-zag lines) extended to paths linking $(S' \setminus A') \cup X$ to S' (by adding the dashed lines). Note that the latter contains paths of length 0 on $S' \setminus A'$.

1. *$v \in C_G(X, A) \cap C_G(X, B)$;*
2. *$C_{G'}(X, A) \cup \{v'\}$ is linked to A and $C_{G'}(X, B) \cup \{v'\}$ is linked to B in G';*
3. *$C_{G'}(X, A) \cup (S \setminus A)' \cup \{v'\}$ is independent in the gammoid (G', S') and $C_{G'}(X, B) \cup (T \setminus B)' \cup \{v'\}$ is independent in the gammoid (G', T').*

Note in particular that our final condition here has no dependence on A or B in the gammoid definitions, only in the choice of "query set". Let M_S (resp. M_T) be the gammoid (G', S') (resp. (G', T')). Now for any vertex $v \in V \setminus (S \cup T)$ and any pair $(A \subseteq S, B \subseteq T)$ with minimum (A, B)-vertex cut X, we find that v is (A, B)-essential in G if and only if $\{v'\}$ extends both $X \cup (S \setminus A)'$ in M_S and $X \cup (T \setminus B)'$ in M_T. This, finally, allows us to identify the set Z we seek, containing all essential vertices.

Theorem 10. *Let a graph $G = (V, E)$ and disjoint sets $S, T \subseteq V$ be given, $|S| = |T| = k$. In randomized polynomial time, with failure probability exponentially small in the input size, we can construct a graph $G' = (V', E')$, where $S, T \subseteq V'$ and $|V'| = \mathcal{O}(k^3)$, such that for every pair of sets $(A \subseteq S, B \subseteq T)$ we have $\lambda_G(A, B) = \lambda_{G'}(A, B)$.*

Proof. We begin with the precise construction, then we argue correctness. Let $M = M_S \oplus M_T \oplus M_k$, where M_S and M_T are as above and $M_k = (V, \binom{V}{\leq k})$ is a k-uniform matroid over V. Note that we can easily produce a linear representation of M in randomized polynomial time. Furthermore, for every $v \in V \setminus (S \cup T)$, let $Y(v)$ consist of the copy of v' in M_S, the copy of v' in M_T, and the copy of v in M_k, and define $\mathcal{Y} = \{Y(v) : v \in V \setminus (S \cup T)\}$. Compute a representative set $\mathcal{Y}^* \subseteq \mathcal{Y}$ of $\mathcal{Y}$ in M using the representative sets lemma, and define $Z = \{v \in V : Y(v) \in \mathcal{Y}^*\}$. Now if possible select a single arbitrary vertex $v \in V \setminus (S \cup T \cup Z)$, bypass v in G, and repeat the procedure, re-computing the matroid M from scratch. When this is no longer possible, let $G' = (V', E')$ be the remaining graph; output this graph and stop. We claim that G' meets all the conditions of the theorem and that the process runs in randomized polynomial time.

For this, we first note that when the procedure halts we have $|V'| = |S| + |T| + |Z|$. Since M has rank $\mathcal{O}(k)$ and $|Y(v)| = 3$ for every $Y(v) \in \mathcal{Y}$, we have $|Z| = \mathcal{O}(k^3)$. Hence $|V'| = \mathcal{O}(k^3)$ and $S, T \subseteq V'$. Furthermore, in each iteration Z can be computed in randomized polynomial time, and there are at most $|V|$ iterations. In particular, since there is only a polynomial number of iterations, the total failure probability remains exponentially small by the union bound. It remains to show that G' has the "cut-mimicking" property that $\lambda_G(A, B) = \lambda_{G'}(A, B)$ for every pair $(A \subseteq S, B \subseteq T)$.

This claim is exactly what is argued in the discussion above. By definition, the result holds if at every step the vertex v that we choose is non-essential (i.e., irrelevant); equivalently, if at every step, the set Z contains all essential vertices. As argued, a vertex $v \in V \setminus (S \cup T)$ is essential if and only if it is (A, B)-essential for some $A \subseteq S$, $B \subseteq T$; by Proposition 4 this is equivalent to $v \in C(X, A) \cap C(X, B)$ for an arbitrary minimum (A, B)-cut X; and by Proposition 5 this is in turn equivalent to $\{v'\}$ extending both $C_{G'}(X, A) \cup (S \setminus A)' \cup \{v'\}$ in M_S and $C_{G'}(X, B) \cup (T \setminus B)' \cup \{v'\}$ in M_T.

If v is (A, B)-essential, we define a set $X(A, B, v)$ consisting of the copy of $C(X, A) \cup (S \setminus A)'$ in M_S, the copy of $C(X, B) \cup (T \setminus B)'$ in M_T, and the copy of $X - v$ in M_k. Then $Y(u)$ extends $X(A, B, v)$ in M if and only if u is (A, B)-essential and $u \notin (X - v)$; clearly, this holds if and only if $u = v$. Hence (in the language of Section 4.2) $X(A, B, v)$ represents a query set in M for which $Y(v)$ is the unique extending tuple in $\mathcal{Y}$; hence $v \in Z$ and we are done. $\square$

Again, it is essential to observe about this proof that the sets $X(A, B, v)$ are never constructed in the algorithm, but are only defined in the analysis phase to prove that the procedure gives the desired result. Indeed, the power of this framework arguably lies exactly in the ability to check $2^k \cdot 2^k$ different "cut questions" (A, B) in time polynomial in both n and k, which would clearly be impossible if the sets $X(A, B, v)$ had to be provided.

We expand on this proof strategy in the next section, where we show a more flexible result for graph cuts, as well as related results for multiway cuts.

5 Cut-covering sets

In this section, we give some of the most central results of the paper, namely the various variants of *cut-covering set* results. In addition to being important in their own right, these results are also heavily used in the kernels in the rest of the paper.

The results are divided into ordinary *two-way cuts* (i.e., standard graph cuts), which also support directed graphs, and results on *multiway cuts*, which only work in the undirected case. All results are phrased for vertex cuts, but the same results naturally apply to edge cuts as well, by standard transformations.

5.1 Covering two-way cuts

We will now prove Theorem 1 and Corollary 1, following the strategy illustrated in Section 4.3. We begin with the basic setup. Let $G = (V, E)$ be a directed graph, and $S, T \subseteq V$ vertex sets, which may overlap, referred to as the *source set* and *sink set*, respectively. For sets $A \subseteq S$, $B \subseteq T$, an (A, B)*-min cut* is an (A, B)-vertex cut X of minimum cardinality, which may overlap A and B. Further, let r be the size of an (S, T)-min cut. We will compute a set $Z \subseteq V$ with $|Z| = \mathcal{O}(|S| \cdot |T| \cdot r)$ such that for any sets A, B as above, Z contains an (A, B)-min cut, in the sense that there is an (A, B)-min cut X with $X \subseteq Z$. In particular, this allows us to compute a set $Z \subseteq V$ with $|Z| = \mathcal{O}((|S| + |T|)^3)$, such that Z contains an (A, B)-min cut for every choice $A \subseteq S$, $B \subseteq T$, although in some applications, the explicit dependency on $|S|$, $|T|$, and r gives a better bound. As announced, we will achieve this via an irrelevant vertex strategy; let us provide the corresponding definitions.

Definition 2. Let G, S, and T be as above. A vertex $v \in V$ is *essential for* (A, B)*-cuts* for $A \subseteq S$ and $B \subseteq T$ if every (A, B)-min cut X has $v \in X$. Further, for the extent of this subsection and when S and T are understood, we say for short that v is *essential* to mean that v is essential for (A, B)-cuts for some $A \subseteq S$ and $B \subseteq T$, and that v is *irrelevant* if there is no $A \subseteq S$ and $B \subseteq T$ such that v is essential for (A, B)-cuts.

We will compute a set $Z \supseteq S \cup T$ that contains every essential vertex in G, then use this set to identify a single vertex $v \in V \setminus Z$ as irrelevant and bypass it. Finally, we will show that repeating this process to exhaustion results in a set Z of vertices that serves as a cut-covering set in the original graph G.

As a first transformation, we ensure that the sets S and T are disjoint, and that S are sources and T sinks in G (i.e., they have in-degree respectively out-degree zero). This can be done without affecting the cut system.

Lemma 9. *Let G, S, and T be as above, and define a graph $G^{\pm}$ from G by adding for every $v \in S$ a new vertex v^- with sole attached arc v^-v, and for every $v \in T$ a new vertex v^+ with sole attached arc vv^+. For $A \subseteq S$ and $B \subseteq T$, let $A' = \{v^- : v \in A\}$ and $B' = \{v^+ : v \in B\}$. Then for every $X \subseteq V$, X is an*

(A,B)-cut in G if and only if it is an (A',B')-cut in $G^{\pm}$. Furthermore, for every minimal (A',B')-cut X' in $G^{\pm}$, we have $(X' \setminus A' \cup B') \subseteq V$.

Proof. Let $u^- \in A'$ and $v^+ \in B'$. Then the set of u^-, v^+-paths in $G^{\pm}$ are exactly the paths $P' = u^- u \dots v v^+$ such that removing the endpoints of P' yields a u,v-path P in G. Hence if $X \subseteq V$ cuts the latter it also cuts the former, and any X which cuts P' while $X \subseteq V$ also cuts the latter. Finally, if $w' \in V(G^{\pm}) \setminus (V \cup A' \cup B')$, then no u^-, v^+-path P' passes through w', hence w' is not used in any minimal vertex cut. $\square$

Henceforth, we simply let $G^{\pm}$, with source set $S^- = \{v^- : v \in S\}$ and sink set $T^+ = \{v^+ : v \in T\}$, replace G, S, and T, that is, we assume that the above properties hold for S and T. It follows from the lemma that once we have a cut-covering set Z for $G^{\pm}$, we may simply use $(Z \cap V(G)) \cup S \cup T$ as cut-covering set for G.

We also define two further graphs G' and G'' where G' is a copy of G with a sink-only copy v' added for every $v \in V$, and G'' is a copy of G which first has all its arcs reversed (replacing every arc $uv \in E$ by an arc vu), then a sink-only copy v' added for every $v \in V$. We can now state our characterization of essential vertices. We begin with a purely graph-theoretical statement.

Lemma 10. *For $A \subseteq S$ and $B \subseteq T$, let C_A be the unique (A,B)-min cut closest to A, and let C_B be the unique (A,B)-min cut in G such that C_B is closest to B in G''; in other words, C_B is also the (B,A)-min cut in G'' that is closest to B. Then a vertex $v \in V \setminus (S \cup T)$ is essential for (A,B)-cuts if and only if $v \in C_A \cap C_B$.*

Proof. Certainly the condition is necessary, as otherwise either C_A or C_B is a witness that v is not essential. On the other hand, assume that $v \in C_A \cap C_B$, and that $X \subseteq V$ is an (A,B)-min cut with $v \notin X$. Clearly, v is not reachable from A in $G-X$ or v cannot reach B in $G-X$ (or both). Assume first the former, i.e., that v is not reachable from A in $G-X$. Let R_X and R_C be the sets of vertices reachable from A in $G-X$ and $G-C_A$ respectively. By minimality of C_A and $v \notin A \subseteq S$ there is an in-neighbor u of v that is reachable from A in $G-C_A$. We must have $u \notin R_X$ or else v would be reachable from A in $G-X$, contradicting our assumption; thus $R_X \cap R_C \subsetneq R_C$. Let $f\colon X \mapsto |N^+(X)|$ be the vertex cut function for directed graphs, where $N^+(X)$ denotes the set of out-neighbors of X and $N^+(X) \cap X = \emptyset$. Then f is submodular[3] and we get $f(R_X \cap R_C) \leq f(R_C) = |C_A|$, contradicting the choice of C_A. By symmetric arguments in G'' the case that v cannot reach B in $G-X$ contradicts the choice of C_B. $\square$

Via Proposition 3, we can translate this into a condition in terms of the gammoids (G',S) and (G'',T).

Lemma 11. *Let $A \subseteq S$ and $B \subseteq T$, and let C_A and C_B be defined as before. Then a vertex $v \in V \setminus (S \cup T)$ is essential for (A,B)-cuts if and only if $\{v'\}$ extends $C_A \cup (S \setminus A)$ in (G',S) and $C_B \cup (T \setminus B)$ in (G'',T).*

Proof. If v is essential then by Lemma 10 we have $v \in C_A \cap C_B$. Since C_A is closest to A and $v \notin A$, by Proposition 3, the set $C_A + v'$ is linked to A in G'. Since $S \supseteq A$ is a set of sources, no vertex of $S \setminus A$ is used for such a linkage. Thus, $C_A \cup (S \setminus A) + v'$ is linked to S in G' and, hence, v' extends $C_A \cup (S \setminus A)$ in (G',S). The same holds for extending $C_B \cup (T \setminus B)$ in (G'',T) (recall that G'' represents the reverse graph of G, so that the roles of sources and sinks are swapped, and C_B is indeed closest to B in G'').

Conversely, assume that $C_A \cup (S \setminus A) + v'$ is linked to S in G' and $C_B \cup (T \setminus B) + v'$ is linked to T in G''. It follows that v' is reachable from S in $G' - C_A \cup (S \setminus A)$ and hence from A in $G' - C_A$; similarly v' is reachable from B in $G'' - C_B$. Since C_A and C_B are cuts, this is only possible if $v \in C_A \cap C_B$. Hence the lemma holds. $\square$

It is not difficult to turn this into a representative sets construction, in line with Section 4.2.

Lemma 12. *Let r denote the size of an (S,T)-min cut in G. Given representations of the gammoids (G',S) and (G'',T) over a common field $\mathbb{F}$ with more than $|V|$ elements, we can in polynomial time compute a set $Z \subseteq V$ of $\mathcal{O}(|S| \cdot |T| \cdot r)$ vertices which includes all essential vertices.*

[3] This is well known, e.g., we can write $f(X) = g(X) - |X|$ where $g(X) = |N^+[X]|$ is the cardinality of the closed out-neighborhood of X. Then g be implemented as a set union cardinality function, cf. Schrijver [70, Chapter 44].

Proof. Create a matroid $M = M_0 \oplus M_1 \oplus M_2$, where M_0 is the uniform matroid of rank r on V, M_1 is the gammoid (G', S), and M_2 is the gammoid (G'', T). Compute a representation of M_0 over $\mathbb{F}$ using an $r \times |V|$ Vandermonde matrix; the size of $\mathbb{F}$ ensures that such a construction exists. Hence we have representations for all three matroids M_0, M_1, M_2, and can compute a representation A of M over the same field $\mathbb{F}$. For a vertex $v \in V$, denote its copy in M_0 by $v(0)$, its two copies in M_1 by $v(1)$ and $v'(1)$, and its two copies in M_2 by $v(2)$ and $v'(2)$. For each $v \in V$ define $Y(v) = \{v(0), v'(1), v'(2)\}$, and let $\mathcal{Y} = \{Y(v) : v \in V \setminus (S \cup T)\}$. Finally, compute a representative set $\mathcal{Y}^*$ for $\mathcal{Y}$ in M, and let $Z = \{v \in V : Y(v) \in \mathcal{Y}^*\} \cup S \cup T$; we claim that Z contains every essential vertex.

For this, let $v \in V \setminus (S \cup T)$ be essential for (A, B)-cuts, for some $A \subseteq S$, $B \subseteq T$, and let C_A (resp. C_B) be the (A, B)-min cut in G that is closest to A in G' (resp. to B in G''). Define a set X in M to contain $C_A - v$ in layer M_0, $C_A \cup (S \setminus A)$ in layer M_1, and $C_B \cup (T \setminus B)$ in layer M_2. By Lemma 11, a vertex u such that $Y(u)$ extends X in layers M_1 and M_2 must be essential for (A, B)-cuts, hence $u \in C_A \cap C_B$; by the layer M_0, the only choice is $u = v$. Hence $Y(u)$ extends X in M if and only if $u = v$, and we conclude $v \in Z$. Since v was arbitrarily chosen, Z must contain all essential vertices.

Finally, the computations can be done in polynomial time according to the representative sets lemma, and the size bound follows by Lemma 4. ☐

We can now complete the proof. We recall the theorem that we are proving.

Theorem 11 (Theorem 1)**.** *Let $G = (V, E)$ be a directed graph and let $S, T \subseteq V$. Let r denote the size of a minimum (S, T)-vertex cut (which may intersect S and T). There exists a set $Z \subseteq V$, $|Z| = \mathcal{O}(|S| \cdot |T| \cdot r)$, such that for any $A \subseteq S$ and $B \subseteq T$, it holds that Z contains a minimum (A, B)-vertex cut. We can find such a set in randomized polynomial time with failure probability $\mathcal{O}(2^{-n})$.*

Proof. The result is obtained via the irrelevant vertex approach outlined in Section 4.3. In each iteration, a single vertex will be bypassed until a graph with $\mathcal{O}(|S| \cdot |T| \cdot r)$ vertices remains; we return the vertex set of that graph as the set Z. We begin with the input graph $G = (V, E)$.

Using Theorem 7 we compute representations of the gammoids $M_1 = (G', S)$ and $M_2 = (G'', T)$. The error chance can be made exponentially small in $n = |V|$ at the cost of a polynomial increase in the used field size. This increases the running time but does not affect the size guarantee for the final set Z. We will need this step less than n times, so by the union bound a failure probability of $\mathcal{O}(n^{-1}2^{-n})$ suffices for an error chance of $\mathcal{O}(2^{-n})$, and we can afford the running time cost for such an error bound. We discuss only the case that all representations are correct.

We use Lemma 12 to compute a set $Z \supseteq S \cup T$ containing all vertices that are essential for (A, B)-min cuts for some $A \subseteq S$ and $B \subseteq T$. If $Z = V$ then we return the set Z. Otherwise, let $v \in V \setminus Z$ and generate the graph $\hat{G}$ by bypassing vertex v: For all $u \in N^-(v)$ and $w \in N^+(v)$ add the arc uw, unless it is already present; then delete vertex v. We prove that this has the desired effect.

Claim 1. *For all $A \subseteq S$ and $B \subseteq T$ the size of (A, B)-min cuts in G and $\hat{G}$ is the same. Moreover, every (A, B)-min cut in $\hat{G}$ is also an (A, B)-min cut in G.*

Proof. Let $A \subseteq S$ and $B \subseteq T$ and let $\hat{X}$ an (A, B)-cut in $\hat{G}$. If $\hat{G}$ was not an (A, B)-cut in G then, since $\hat{G}$ is a supergraph of $G - v$, there must be an A, B-path in $G - \hat{X}$ using vertex v. By construction of $\hat{G}$ this directly yields an A, B-path in $\hat{G} - \hat{X}$; a contradiction.

Let X an (A, B)-min cut in G; by Lemma 12 vertex $v \notin Z$ is not essential and we can choose X such that it avoids v. If X was not an (A, B)-cut in $\hat{G}$ then there would have to be an A, B-path in $\hat{G} - X$ using one of the newly introduced arcs uw for $u \in N^-(v)$ and $w \in N^+(v)$. This directly corresponds to an A, B-path through in $G - X$ that contains v, using $v \notin X$; a contradiction.

Thus, (A, B)-min cut sizes in G and $\hat{G}$ must be the same. The moreover part now follows from the first paragraph. ☐

It follows that when the algorithm terminates the final graph $\hat{G}$ has the same (A, B)-min cut sizes as the initial graph and that each (A, B)-min cut in $\hat{G}$ is also an (A, B)-min cut in the initial graph. Thus, the set $Z = V(\hat{G})$ fulfills the required properties. Note that the size bound follows directly from Lemma 12. Clearly, each iteration can be performed in polynomial time and there are at most n iterations. This completes the proof. $\square$

The following was an explicit open question.[4]

Corollary 4. *Let $G = (V, E)$ be a directed graph, and let $S, T \subseteq V$. Let $r \leq \min(|S|, |T|)$ denote the cardinality of an (S, T)-min cut in G. In randomized polynomial time we can compute a graph $G' = (V', E')$ with $S \cup T \subseteq V' \subseteq V$ such that $|V'| = \mathcal{O}(|S| \cdot |T| \cdot r)$ such that (G, S) and (G', S) define the same gammoid when restricted to T.*

Proof. Compute the cut-covering set $Z \supseteq S \cup T$ as in Theorem 1, and let G' be obtained by bypassing all vertices in $V \setminus Z$ (also known as taking the torso to Z). Then clearly, assuming the process succeeded, for every set $X \subseteq T$ the (S, X)-min cuts in G and in G' are of the same cardinality. $\square$

It is straightforward to extend Theorem 1 to the following more versatile result.

Corollary 5 (Corollary 1)**.** *Let $G = (V, E)$ be a directed graph, and $W \subseteq V$ a set of terminals. We can identify, in randomized polynomial time, a set $Z \supseteq W$ of $\mathcal{O}(|W|^3)$ vertices such that for any partition $W = A \cup B \cup R \cup Q$, a minimum (A, B)-vertex cut in $G - R$ is contained in Z.*

Proof. Apply Theorem 1 to G and $S = T = W$. Let us check that the returned set $Z \supseteq S \cup T = W$ contains the required cuts. Fix any partition $W = A \cup B \cup R \cup Q$ and define $A' = A \cup R \subseteq S$ and $B' = B \cup R \subseteq T$. By Theorem 1, there is an (A', B')-min cut $X \subseteq Z$. Clearly, $R = A' \cap B' \subseteq X$. Thus, $X \setminus R$ is an (A, B)-min cut in $G - R$ that is a subset of Z. Running time and error bound follow from Theorem 1. $\square$

5.2 Covering s-way cuts

We now derive similar consequences for cuts partitioning a set of terminals into at most s components. This will be used for kernelization consequences in Section 7. Throughout this section, $G = (V, E)$ is an undirected graph, $T \subseteq V$ a set of terminals, and $s \in \mathbb{N}$ a constant. We wish to preserve a minimum multiway cut for every way to partition T into at most s parts, as defined below. We begin with a result where T is partitioned fully, then extend to consider partitions with deletions as in Corollary 1.

Definition 3. Let $\mathcal{T} = (T_1, \ldots, T_s)$ be a partition of T. A *multiway cut of* $\mathcal{T}$ is a set of vertices $X \subseteq V$ such that for every pair $t_i, t_j \in T$ belonging to different parts of $\mathcal{T}$, $G - X$ has no path from t_i to t_j; a *minimum multiway cut of* $\mathcal{T}$ is one which in addition has minimum cardinality. We say that a set $Z \subseteq V$ *covers all s-way cuts of T (in G)* if, for every such partition $\mathcal{T}$, there is a minimum multiway cut X of $\mathcal{T}$ with $X \subseteq Z$. Note that X may overlap T in all the above. Furthermore, for $v \in V$, we say that v is *essential for partition* $\mathcal{T}$ if every minimum multiway cut X of $\mathcal{T}$ has $v \in X$, and *irrelevant (for simple partitions)* if v is not essential for any partition $\mathcal{T}$ of T.

We will prove Theorem 2, showing that there is a set $Z \subseteq V$ with $|Z| = \mathcal{O}(|T|^{s+1})$ which covers all s-way cuts of T; the process will follow the outline sketched in Section 4.3. Let us make this more formal. Let G' be the directed graph obtained from G by adding a sink-only copy v' for every $v \in V \setminus T$.

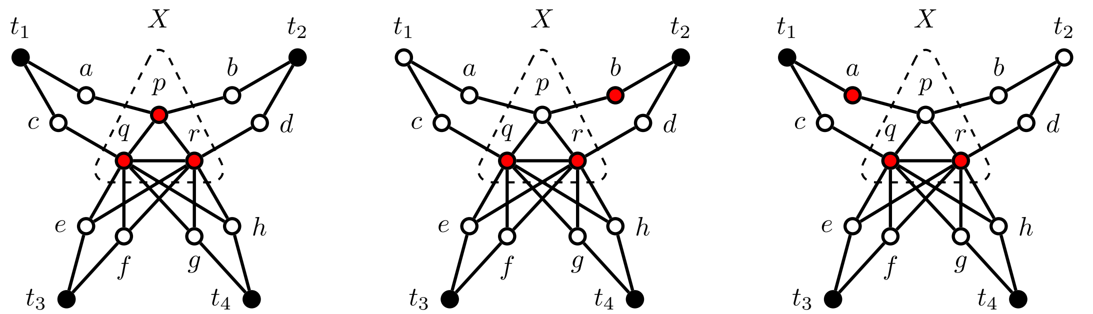
Definition 4. Let $\mathcal{T} = (T_1, \ldots, T_s)$ be a partition of T, and $X \subseteq V$ a minimum multiway cut of $\mathcal{T}$. We say that $v \in V \setminus T$ is *central for X (w.r.t. $\mathcal{T}$)* if for every $i \in [s]$ we have $v \in C(X, T \setminus T_i)$.[5]

See Figure 3 for an illustration of centrality. We begin by a few properties showing that central vertices are a good proxy for essential vertices.

Lemma 13. *Let $\mathcal{T} = (T_1, \ldots, T_s)$ be a partition of T and let X be a multiway cut of $\mathcal{T}$. The following hold.*

[4] See `http://lemon.cs.elte.hu/egres/open/Small_representations_of_gammoids`.

[5] The notion *central vertex* corresponds roughly to the term *highly reachable* used in the previous version of this paper [47]. We prefer the term central vertex here because the centrality condition maps more transparently to solutions for cut problems.



(a) The graph G with X marked (b) The set $C(X, T - t_1)$ (c) The set $C(X, T - t_2)$

Figure 3: Illustration of centrality (see Definition 4). The set $X = \{p, q, r\}$ is a solution to the multiway cut instance with terminals $T = \{t_1, t_2, t_3, t_4\}$. The vertices q and r are central, but p is not.

1. *For every $i \in [s]$, the set $X_i := C(X, T \setminus T_i)$ is a multiway cut of $\mathcal{T}$, and $|X_i| \leq |X|$.*

2. *For every $v \in V \setminus T$, if v is central for X then $v \in X$.*

In particular, if $v \in V \setminus T$ is essential for partition $\mathcal{T}$, then for any minimum multiway cut X of $\mathcal{T}$, v is central for X.

Proof. For the first part, let P be a path in G from $t_a \in T_a$ to $t_b \in T_b$, for some $a \neq b$, and assume w.l.o.g. that $a \neq i$. Then P contains a path from $t_a \in (T \setminus T_i)$ to X, hence X_i intersects P. Thus X_i is a multiway cut of $\mathcal{T}$. For the second condition, we have $|X_i| \leq |X|$ since X_i is a $(T \setminus T_i, X)$-min cut.

For the second part, if $v \notin X$, then v is reachable from at most one part T_i in $G - X$. Assume that v is reachable from T_i in $G - X$. Then $v \notin X_i = C(X, T \setminus T_i)$: Otherwise, there would need to be an $X, T \setminus T_i$-path P whose only intersection with X_i is the vertex v. Clearly, because v is reachable from T_i and X is a multiway cut, between the last occurrence of v in P and a vertex of $T \setminus T_i$ there must lie a vertex x of X; this yields a subpath from x to $T \setminus T_i$ that does not intersect X_i; a contradiction. (If v is reachable from no part T_i in $G - X$, we may apply the same argument to an arbitrary part T_i.)

Finally, if $v \in V \setminus T$ is essential, then every minimum multiway cut contains v, hence in particular $v \in X_i$ for every $i \in [s]$ and every minimum multiway cut X of $\mathcal{T}$ because $|X_i| \leq |X|$ implies that each X_i is itself a minimum multiway cut. ☐

We now translate the notion of central vertices into a condition suitable for representative sets.

Lemma 14. *Let $\mathcal{T} = (T_1, \ldots, T_s)$ be a partition of T and X a minimum multiway cut of $\mathcal{T}$. A vertex $v \in V \setminus T$ is central for X if and only if $\{v'\}$ extends $X \cup T_i$ in (G', T) for every $i \in [s]$.*

Proof. For $i \in [s]$, let $X_i = C(X, T \setminus T_i)$. On the one hand, if $v \in X_i$ then (since X_i is a closest set) $X_i + v'$ is linked to $T \setminus T_i$ by Proposition 3(1). By Lemma 13(1) X_i is a multiway cut with $|X_i| \leq |X|$, and since X is a minimum multiway cut we have $|X| = |X_i|$. Since X_i is a $(T \setminus T_i, X)$-cut it also follows that X is linked to X_i. We can combine the two path systems to a set of paths witnessing that $X + v'$ is linked to $T \setminus T_i$ in G'. Furthermore, since X is a multiway cut, none of these paths can intersect a vertex of $T_i \setminus X$; hence we may add length-zero paths from $T_i \setminus X$ to itself, showing that $(X \cup T_i) + v'$ is linked to T in G'.

On the other hand, assume that $(X \cup T_i) + v'$ is linked to T for each $i \in [s]$. Note that $C(X \cup T_i, T) = X_i \cup T_i$ for each $i \in [s]$: X_i is closest to $T \setminus T_i$, and every $(X \cup T_i, T)$-cut contains T_i. Hence by Proposition 3(2), v' is reachable from $T \setminus T_i$ in $G - X_i$ for every $i \in [s]$. Assume for a contradiction that $v \notin X_i$ for some $i \in [s]$. Then v is reachable from $T \setminus T_i$ in $G - X_i$, say from a terminal $t \in T_j$ for some $j \neq i$. Since X_i separates X from $T \setminus T_i$, this path must be disjoint from X and v is connected to T_j in $G - X$. But then $X \cup T_j$ separates v from T in G since X is a multiway cut, contradicting that $(X \cup T_j) + v'$ is linked to T. ☐

We can now complete the proof.

Theorem 12 (Theorem 2)**.** *Let $G = (V, E)$ be an undirected graph and $T \subseteq V$. For any s, there exists a set $Z \subseteq V$, $|Z| = \mathcal{O}(|T|^{s+1})$, such that for any partition $\mathcal{T} = (T_1, \ldots, T_s)$ of T, there is a minimum multiway cut X of $\mathcal{T}$ with $X \subseteq Z$. We can find such a set in randomized polynomial time with failure probability $\mathcal{O}(2^{-n})$.*

Proof. First we show how to compute a set Z which contains every vertex central for X, for every minimum multiway cut X of every partition $\mathcal{T}$, then we use this to compute a cut-covering set in the original graph G.

For the first part, we proceed as in Lemma 12. Let $M = M_0 \oplus \ldots \oplus M_s$, where M_0 is the uniform matroid on V of rank $|T|$ and each M_i, $i \in [s]$, is the gammoid (G', T). For a vertex $v \in V(G')$, let $v(i)$ denote its copy in M_i. We define $Y(v) = \{v(0), v'(1), \ldots, v'(s)\}$ for each $v \in V \setminus T$, and let $\mathcal{Y} = \{Y(v) : v \in V \setminus T\}$. Compute a representation of M, and a representative set $\mathcal{Y}^*$ of $\mathcal{Y}$ in M, and let $Z = \{v \in V : Y(v) \in \mathcal{Y}^*\}$. We claim that Z contains all vertices as claimed.

To show this, let $\mathcal{T} = (T_1, \ldots, T_s)$ be a partition of T, X a minimum multiway cut of $\mathcal{T}$, and v a central vertex for X. Define a set S in M consisting of $X - v$ in M_0 and $X \cup T_i$ in M_i for each $i \in [s]$. By Lemmas 14 and 13, $Y(v)$ extends S, and for every $u \in V$ such that $Y(u)$ extends S we have $u \in X$. But then $u = v$ by the use of layer M_0. We conclude that $Y(u)$ extends S if and only if $u = v$, hence $v \in Z$. This finishes the first part.

To complete the proof, we follow the same procedure as in Theorem 1. If $Z \cup T = V$, then we are done. Otherwise let $v \in V \setminus (Z \cup T)$ be arbitrarily chosen, bypass v in G, and repeat the whole procedure until it terminates. Let the final resulting graph be $\hat{G} = (\hat{V}, \hat{E})$, and note that $T \subseteq \hat{V} \subseteq V$, and that $|\hat{V}| = \mathcal{O}(|T|^{s+1})$ (since otherwise we could still find a vertex v to bypass). We claim that the set $\hat{V}$ is the cut-covering set we seek.

To show this, let $\mathcal{T} = (T_1, \ldots, T_s)$ be an arbitrary partition of T, and let X be a minimum multiway cut for $\mathcal{T}$ in $\hat{G}$. By Proposition 2, X is also a multiway cut in G. We claim that X is also of minimum cardinality: indeed, by design, the bypassed vertex $v \in V$ is irrelevant in G. Therefore, there is a minimum multiway cut X' for $\mathcal{T}$ in G with $v \notin X'$, which is preserved by the bypass step. By induction, the size of a minimum multiway cut is preserved by the entire reduction process. Therefore X is a minimum multiway cut of $\mathcal{T}$ in G. Since $\mathcal{T}$ was chosen arbitrarily, the result follows. ☐

We now extend the results as promised, to cover subpartitions with optional vertex deletion. Let a *generalized s-partition* $\mathcal{T} = (T_0, T_1, \ldots, T_s, T_X)$ of T be a partition of T into $s + 2$ parts (some possibly empty), with two distinguished parts T_0, T_X, referred to as the *free* and the *deleted* part of $\mathcal{T}$, respectively. Let $T' = T_1 \cup \ldots \cup T_s$. A *multiway cut of* $\mathcal{T}$ is a multiway cut in $G - T_X$ of the partition $\mathcal{T}' = (T_1, \ldots, T_s)$ of T', i.e., vertices in T_X are deleted and vertices in T_0 are unspecified. As before, a *minimum multiway cut of* $\mathcal{T}$ is one of minimum cardinality. We show that with a slight variation of the tools, we can cover minimum multiway cuts even in generalized partitions.

Theorem 13. *Let $G = (V, E)$ be an undirected graph with a set $T \subseteq V$ of terminal vertices, and let $s \in \mathbb{N}$ be a constant. There is a set $Z \subseteq V$ with $|Z| = \mathcal{O}(|T|^{s+1})$ such that Z contains a minimum multiway cut of every generalized s-partition $\mathcal{T}$ of T, and we can compute such a set in randomized polynomial time with failure probability $\mathcal{O}(2^{-n})$.*

Proof. Assume $s \geq 3$, as otherwise the result follows from Corollary 1. Construct a graph G' by adding three pendant vertices t', t'', t''' to every terminal $t \in T$ (i.e., with edges $\{t', t\}$, $\{t'', t\}$, and $\{t''', t\}$). Let G' be the resulting (undirected) graph and let $T' = \{t', t'', t''' : t \in T\}$ be the new set of terminals. Finally, compute an s-way cut-covering set for G' and T' as in Theorem 2. Let $Z' \subseteq V(G')$ be the resulting set; we argue that $Z := (Z' \cap V) \cup T$ is the sought-after set for G.

For that, let $\mathcal{T} = (T_0, T_1, \ldots, T_s, T_X)$ be a generalized s-partition of T, with special sets T_X and T_0, and let $X \subseteq V$ be a minimum multiway cut of $\mathcal{T}$. We define an s-partition $\mathcal{T}' = (T'_1, \ldots, T'_s)$ of T' as follows. For every $t \in T_i$, $i \in [s]$, we place all three terminals t', t'', t''' in T'_i. For every $t \in T_X$, we place t' in T_1, t''

in T_2 and t''' in T_3. Finally, for $t \in T_0$, if t is reachable from a terminal in T_i in $G \setminus X$ for some $i \in [s]$, then the terminals t', t'', t''' are all placed in T'_i; otherwise they are placed in T'_1. This completes the description of $\mathcal{T}'$. Let $X' \subseteq Z'$ be a minimum multiway cut of $\mathcal{T}'$ in G'. We show that X' contains a minimum multiway cut of $\mathcal{T}$ in G.

First of all, we argue that for every $t \in T_X$ we have $t \in X'$. Indeed, if $t \notin X'$ then X' must contain at least two vertices out of t', t'', t''', and replacing these two by t would produce a smaller multiway cut. Second, let $t_i \in T_i$ and $t_j \in T_j$ be arbitrary terminals in different parts of $\mathcal{T}$. Again by the optimality of X', we observe that X' does not contain all three terminals t', t'', t''' for any $t \in \{t_i, t_j\}$; hence X' intersects every path from t_i to t_j in G. This shows that $X' \setminus T_X$ is a multiway cut of $(T_1, \ldots, T_s)$ in $G - T_X$, hence by definition $X' \setminus T_X$ is a multiway cut of $\mathcal{T}$ in G.

To argue the optimality of $X' \setminus T_X$, we show that $X \cup T_X$ is a multiway cut of $\mathcal{T}'$ in G'. Indeed, let $t_i \in T'_i$ and $t_j \in T'_j$, with $i \neq j$. If t_i and t_j represent the same terminal $t \in T$, then $t \in T_X$. Otherwise, t_i and t_j represent distinct terminals in T which are separated by $X \cup T_X$ in G, using that terminals from T_0 are placed in T_i if they are reachable from any $t \in T_i$ in $G - X$. In both cases, $X \cup T_X$ separates t_i from t_j in G', showing that $X \cup T_X$ is a valid multiway cut of $\mathcal{T}'$ in G'. Now $|X'| \leq |X \cup T_X|$ by the optimality of X', and we have $|X'| = |X' \setminus T_X| + |T_X| \leq |X \cup T_X| = |X| + |T_X|$, implying $|X' \setminus T_X| \leq |X|$, as required. This confirms that $X' \setminus T_X$ is an optimal multiway cut for $\mathcal{T}$ in G.

Finally, the cardinality of Z, the running time, and the success probability follow from Theorem 2. $\square$

6 Almost 2-SAT and related problems

In this section we give randomized polynomial kernels for ALMOST 2-SAT and several related problems. Our first target is a problem that we call DIGRAPH PAIR CUT, which captures the iterative compression version of ALMOST 2-SAT (see Section 6.1). Our kernelization for this problem uses a variant of the constraint reduction approach described in Section 4.1 and then applies the two-way cut-covering result from Section 5.1. The result for ALMOST 2-SAT then follows by a polynomial parameter transformation to DIGRAPH PAIR CUT via its compression version, bootstrapped with an approximate solution (see Section 6.2). The further results follow by (mostly) known reductions to ALMOST 2-SAT (see Section 6.3).

In order to improve the size of the kernel, we also give a new approximation result for ALMOST 2-SAT (see Section 6.2). Previous results have shown an $\mathcal{O}(\sqrt{\log n})$-approximation [3], which we used in a previous version of this paper [47] combined with an FPT algorithm to give a solution of size $\mathcal{O}(\mathrm{OPT}^{1.5})$, where OPT denotes the size of an optimal solution; furthermore, a result of Even et al. [24] on SYMMETRIC MULTICUT in directed graphs can be used to provide an $\mathcal{O}(\log \mathrm{OPT} \log\log \mathrm{OPT})$-approximation. We show that the kernel and approximation can be combined to give a tighter approximation guarantee of $\mathcal{O}(\sqrt{\log \mathrm{OPT}})$, implying a kernel for ALMOST 2-SAT with $\mathcal{O}(k^4 \log^2 k)$ variables.

For a 2-CNF formula $\mathcal{F}$, let a *deletion set* X be a set of variables of $\mathcal{F}$ such that removing every clause containing a variable in X leaves a satisfiable formula. We define the problem ALMOST 2-SAT as follows. (Note that the problem may alternatively be defined by deleting at most k *clauses*; these two forms are equivalent under simple transformations, cf. [54].)

ALMOST 2-SAT(k)	**Parameter:** k.
Input: A formula $\mathcal{F}$ in 2-CNF and an integer $k \in \mathbb{N}$.	
Question: Is there a deletion set X for $\mathcal{F}$ of size at most k?	

For technical convenience we assume that formulas $\mathcal{F}$ consist only of 2-clauses, i.e., there are no singletons (x) resp. $(\neg x)$. For both variable- and clause-deletion variant this can be easily achieved by replacing, e.g., (x) by $(x \vee y) \wedge (x \vee \neg y)$ where y is a new variable. The deletion cost stays the same for both variants. We will tacitly assume this form of replacement when (for ease of reading) we say that we add (x) or $(\neg x)$ to a clause.

6.1 A polynomial kernelization for Digraph Pair Cut(k)

In this section we provide a randomized polynomial kernelization for DIGRAPH PAIR CUT(k); the problem is formally defined as follows.

DIGRAPH PAIR CUT(k) — DPC(k)	**Parameter:** k.
Input: A directed graph $D = (V, A)$ with source vertex $s \in V$, a set of pairs $\mathcal{P} \subseteq \binom{V}{2}$, and an integer $k \in \mathbb{N}$.	
Question: Is there a set $X \subseteq V \setminus \{s\}$ with $\lvert X\rvert \leq k$ such that no pair in $\mathcal{P}$ is reachable in $D - X$, i.e., such that for each $\{u, v\} \in \mathcal{P}$ at least one of u and v is not reachable from s in $D - X$?	

DIGRAPH PAIR CUT can be seen as a cut-based generalization of VERTEX COVER. Specifically, if the graph D is an n-point star with s in the center, then the input is equivalent to a VERTEX COVER instance with one edge for every pair in $\mathcal{P}$.

Another view is as follows. Let a *$\mathcal{P}$-transversal* be a set $T \subseteq \bigcup \mathcal{P}$ that intersects every pair $P \in \mathcal{P}$. The problem DIGRAPH PAIR CUT can then be phrased as seeking a $\mathcal{P}$-transversal of cost $f(T) \leq k$, where $f(T)$ is the size of a minimum (s, T)-cut in D. Since cut functions are submodular, this shows that DIGRAPH PAIR CUT is a special case of the generic problem SUBMODULAR VERTEX COVER. More specifically, Lovász [51] refers to the combination of a graph $G = (V, E)$ and a matroid M on ground set V as a *pregeometric graph*, and refers to the problem of finding a vertex cover for G of minimum rank in M as the *covering problem* for pregeometric graphs. The same problem is referred to by Meesum et al. [56] as RANK VERTEX COVER. Taking the set of pairs $\mathcal{P}$ as the edges of a graph, and the gammoid on $W = \bigcup \mathcal{P}$ as the matroid, we see that DIGRAPH PAIR CUT is a special case of RANK VERTEX COVER. Meesum et al. [56] built on the work of Lovász to give an alternative polynomial instance compression for VERTEX COVER ABOVE LP, and thereby also for ALMOST 2-SAT.

The following theorem shows that DIGRAPH PAIR CUT can be solved almost as efficiently as VERTEX COVER, using the following observation.

Observation 1. *We can safely restrict our attention to solutions X that are* closest *to s: If X is a solution then every (s, X)-cut X' of size at most $\lvert X\rvert$ must also be a solution; if X is not closest then such a closest cut X' exists.*

Theorem 14. *The* DIGRAPH PAIR CUT *problem can be solved in time $\mathcal{O}^*(2^k)$.*

Proof. Let a directed graph $D = (V, A)$, a vertex $s \in V$, a set of pairs $\mathcal{P}$, and an integer k be given. Based on Observation 1 the algorithm will seek a solution that is closest to s. The algorithm solves the more general problem of checking for a $\mathcal{P}$-transversal of cost at most k which contains T, for a given vertex set T. This is equivalent to DIGRAPH PAIR CUT for $T = \emptyset$.

1. Initialize by creating copies $S = \{s_1, \ldots, s_{k+1}\}$ of s (to make s undeletable), and let $T = \emptyset$.
2. Let C be the min-(S, T)-cut closest to S. If $\lvert C\rvert > k$, **reject** the instance. If no pair is reachable in $D - C$, **accept** the instance.
3. Otherwise, let $P = \{u, v\}$ be a pair reachable in $D - C$. Branch two ways, adding either u or v to T and return to Step 2.

Since every recursion in Step 3 increases the min-cut size λ, at most k recursion steps are taken, and the algorithm runs in time $\mathcal{O}^*(2^k)$ as promised. □

We have already discussed in Section 4.1 the key ideas for how to reduce the number of pairs in $\mathcal{P}$. This will form the first half of the kernelization; the second consists of an application of the two-way cut-covering result of Theorem 1. To adhere to the standard gammoid definition where each source vertex has capacity one, gammoid tools will be applied to the graph D_s obtained from D by using a set S of $k + 1$ copies of s as sources.

For closest solutions X, in fact for any closest set X, Proposition 3 tells us that reachability of v from s in $D - X$ is equivalent to independence of $X \cup \{v\}$. Thus, the vertices p, q of some pair $P \in \mathcal{P}$ are reachable from s relative to a (proposed) solution X if and only if both $X \cup \{p\}$ and $X \cup \{q\}$ are independent in the gammoid (D_s, S). As discussed earlier, we have no reason to assume that in general also $X \cup \{p, q\}$ would be independent, and instead we will use independence in a matroid that consists of two disjoint copies of the gammoid (D_s, S).

Lemma 15. *There is a randomized algorithm that given an instance $(D, s, \mathcal{P}, k)$ of* DPC(k) *takes time polynomial in the input size and, with failure probability exponentially small in $|D|$, returns a set $\mathcal{P}^* \subseteq \mathcal{P}$ of size $\mathcal{O}(k^2)$, such that for any set $X \subseteq V \setminus \{s\}$ closest to s of at most k vertices, the graph $D - X$ contains a reachable pair $P \in \mathcal{P}$ if and only if it contains a reachable pair $P^* \in \mathcal{P}^*$.*

Proof. Split s into $k+1$ copies, and let S contain these vertices; let D_s denote the obtained graph. Form a matroid M by taking two disjoint copies of the gammoid (D_s, S); let $v(1)$ and $v(2)$ be the copies of a vertex $v \in V$ in the first resp. the second part of this matroid. Note that M has rank $2k+2$. Now, form a set $\mathcal{Y}$ by taking, for every $\{u, v\} \in \mathcal{P}$, the pair $\{u(1), v(2)\}$. Use the representative sets lemma to compute a representative subset $\mathcal{Y}^*$ of $\mathcal{Y}$, and let $\mathcal{P}^*$ contain all pairs that occur in $\mathcal{Y}^*$. A representation of M can be computed in randomized polynomial time, with failure probability as requested, by Theorem 7, and the size of $\mathcal{P}^*$ and the running time follow from the representative sets lemma.

Let $X \subseteq V \setminus \{s\}$ be any closest set of size at most k and let $X' = \{x(1), x(2) \mid x \in X\}$. Note that X is also closest to S in D_s. Assume that $D - X$ contains a reachable pair $P = \{u, v\} \in \mathcal{P}$. It follows that P is also reachable from S in $D_s - X$. By Proposition 3 and Lemma 3, we have that $\{u(1), v(2)\} \in \mathcal{Y}$ extends X', and since $\mathcal{Y}^*$ is a representative set, some pair in $\mathcal{Y}^*$ extends X' as well. This represents some reachable pair $P^* \in \mathcal{P}^*$ in $D_s - X$. Since $X \subseteq V \setminus \{s\}$ the pair P^* is also reachable in $D - X$. The converse is trivial since $\mathcal{P}^* \subseteq \mathcal{P}$. ☐

Note that the bound on $|\mathcal{P}^*|$ is tight even in the case of VERTEX COVER, where $\mathcal{O}(k^2)$ edges is optimal [19].

Henceforth, we may assume that instances $(D, s, \mathcal{P}, k)$ of DPC(k) have $|\mathcal{P}| \in \mathcal{O}(k^2)$. We can use this to describe solutions for DPC(k) as optimal (S, T)-cuts where $T \subseteq \bigcup \mathcal{P}$ is a $\mathcal{P}$-transversal. This perspective will allow us, later, to directly apply the two-way cut-covering result and ensure that some optimal solution is preserved.

Lemma 16. *Let $(D, s, \mathcal{P}, k)$ be an instance of* DPC(k) *with $D = (V, A)$, and let D_s be obtained by taking a set S of $k+1$ copies of s. If $X \subseteq V \setminus \{s\}$ separates each pair of $\mathcal{P}$ from s then there is a $\mathcal{P}$-transversal $T \subseteq \bigcup \mathcal{P}$ such that X is an (S, T)-cut in D_s. Conversely, if $X \subseteq (V \setminus \{s\}) \cup S$ with $|X| \leq k$ is an (S, T)-cut in D_s for a $\mathcal{P}$-transversal $T \subseteq \bigcup \mathcal{P}$ then in $D - (X \setminus S)$ no pair in $\mathcal{P}$ is reachable from s.*

Proof. Let $X \subseteq V \setminus \{s\}$ be such that in $D - X$ no pair of $\mathcal{P}$ is reachable from s. Pick a $\mathcal{P}$-transversal T by selecting for each pair $P \in \mathcal{P}$ a vertex $v \in P$ that is not reachable from s in $D - X$; clearly such a choice exists for every P. Observe that no copy of s in D_s can trace a path to a vertex $v \in T$ in $D_s - X$ since that would give a path from s to v in $D - X$. Thus, X is indeed an (S, T)-cut in D_s for a $\mathcal{P}$-transversal T.

Conversely, fix any $\mathcal{P}$-transversal T and let $X \subseteq (V \setminus \{s\}) \cup S$ be an (S, T)-cut of size at most k in D_s. Since $|X| \leq k$ we can pick some vertex $s' \in S \setminus X$ that is present in $D_s - X$. Since X separates S from T there cannot be a path from s' to T in $D_s - X$. It follows directly that $X \setminus S$ also separates s from T in $D - X$. This implies that no pair $P \in \mathcal{P}$ is reachable from s in $D - (X \setminus S) = D - X$. ☐

Using the two-way cut-covering theorem we obtain a kernelization for DIGRAPH PAIR CUT(k). Our kernelization for ALMOST 2-SAT(k) uses a reduction to DIGRAPH PAIR CUT(k) but we obtain a better size bound by a tailored argument rather than using the present kernelization as a black box (as in the previous version).

Theorem 15 (Theorem 3)**.** DIGRAPH PAIR CUT(k) *admits a randomized polynomial kernelization that on input $(D, s, \mathcal{P}, k)$ returns an equivalent instance with $\mathcal{O}(k^4)$ vertices. The error probability is exponentially small in the input size.*

Proof. Given $(D, s, \mathcal{P}, k)$ with $D = (V, A)$ the kernelization proceeds as follows.

- Reduce the number of pairs in $\mathcal{P}$ to $\mathcal{O}(k^2)$ using Lemma 15.
- Apply Theorem 1 on D_s, S, and $W = \bigcup \mathcal{P}$ to compute a set $Z \subseteq V$ of size at most $\mathcal{O}(|S| \cdot |W| \cdot (k+1)) = \mathcal{O}(k^4)$ that for every $A \subseteq S$, $B \subseteq W$ contains the vertices of a minimum (A, B)-cut.
- Apply the torso operation with the vertex set $Z' = \{s\} \cup W \cup (Z \setminus S)$ in D:
  - Add an arc (u, v) for $u, v \in Z'$ if there is a directed path from u to v with no internal vertex from Z'.
  - Delete all vertices of $V \setminus Z'$ from D to obtain D'.

  Return $(D', s, \mathcal{P}, k)$ as the kernelized instance.

The number of vertices is dominated by the bound of $\mathcal{O}(k^4)$ on the size of Z by Theorem 1. Each step can be performed in polynomial time with failure probability at most exponentially small in the input size.

Correctness. By Lemma 15 the reduction of $|\mathcal{P}|$ to $\mathcal{O}(k^2)$ is correct (up to the exponentially small error probability), and preserves closest solutions of all sizes $t \leq k$. Thus, taking the torso operation on any vertex set containing s and all remaining pairs can only err on making the instance negative. (Recall that the torso operation captures making vertices undeletable, which shrinks the space of feasible solutions.) It remains to show that positive instances with $\mathcal{O}(k^2)$ pairs are correctly handled.

Let $(D, s, \mathcal{P}, k)$ be a **yes**-instance of DPC(k) with $|\mathcal{P}| = \mathcal{O}(k^2)$, and let $X \subseteq V \setminus \{s\}$ of size at most k such that no pair in $\mathcal{P}$ is reachable from s in $D - X$. Pick a $\mathcal{P}$-transversal $T_X \subseteq W = \bigcup \mathcal{P}$ such that X is an (S, T_X)-cut in D_s. This upper bounds the size of *minimum* (S, T_X)-cuts by $|X|$. Now by Theorem 1 there is a minimum (S, T_X)-cut X^* contained in Z, i.e. $X^* \subseteq Z$. Clearly, $|X^*| \leq |X| \leq k$. Observe also that $X^* \cap S = \emptyset$ since deleting S only partially has no effect and $|S| > |X^*|$. We claim that X^* is a solution for the kernelized instance $(D', s, \mathcal{P}, k)$.

Assume for contradiction that in $D' - X^*$ some pair $P \in \mathcal{P}$ is reachable from s. Let $v \in P \cap T_X$ and consider a path from s to v in $D' - X^*$. By definition of the torso operation this directly corresponds to a path in D from s to v with no internal vertex from X^* since arcs introduced by the operation can be replaced by paths with internal vertices from $V \setminus Z'$, which is disjoint from $X^* \subseteq Z'$. This directly gives a path from S to $v \in T_X$ in D_s, contradicting the choice of X^* as (S, T_X)-cut. Thus, X^* is indeed a solution for $(D', s, \mathcal{P}, k)$. $\square$

We note that a generalization of the results of this section from pairs to q-tuples still holds. Consider the generalization of DIGRAPH PAIR CUT to the problem DIGRAPH q-TUPLE CUT, where the input contains a set Q of q-tuples over V instead of a set of pairs $\mathcal{P}$, and where the task is to separate at least one member of every q-tuple from s, at cost at most k. Then the natural generalizations of the above results allow us to reduce Q to a set of $\mathcal{O}(k^q)$ representative q-tuples, and give a polynomial kernel for DIGRAPH q-TUPLE CUT of $\mathcal{O}(k^{q+2})$ vertices. Again, note that a bound of $\mathcal{O}(k^q)$ on the size of Q in the kernel is tight for q-tuples, by a straightforward reduction from q-HITTING SET, which by Dell and van Melkebeek [19] has no kernel of total size of $\mathcal{O}(k^{q-\epsilon})$ for any $\epsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{coNP}$/poly.

6.2 A polynomial kernelization for Almost 2-SAT(k)

We will now give a randomized polynomial kernel for ALMOST 2-SAT(k), by reducing the iterative compression form of the problem to DIGRAPH PAIR CUT(k). By bootstrapping the compression step with an approximate solution of appropriate size, we get a polynomial kernel. Let us begin with explicitly defining the iterative compression form of ALMOST 2-SAT.

ALMOST 2-SAT COMPRESSION$(\lvert X \rvert)$	**Parameter:** $\lvert X \rvert$.
Input: A 2-CNF formula $\mathcal{F}$, an integer k, and a deletion set X for $\mathcal{F}$.	
Question: Is there a deletion set X' for $\mathcal{F}$ of size at most k?	

For a 2-CNF formula $\mathcal{F}$ and a set of variables S occurring in $\mathcal{F}$, let $\mathcal{F} - S$ denote the result of removing every clause that intersects S from $\mathcal{F}$. The reduction from ALMOST 2-SAT COMPRESSION$(|X|)$ to DIGRAPH

PAIR CUT(k) goes as follows. Let $(\mathcal{F}, k, X)$ be an instance of ALMOST 2-SAT COMPRESSION($|X|$) on variable set V. By negating variables in $\mathcal{F}$, we ensure that $\mathcal{F} - X$ is zero-valid, i.e., satisfied by the assignment which sets all variables to false. (To do so, take any satisfying assignment of $\mathcal{F} - X$ and negate all variables that are set to 1.) In particular, this means that there are no positive clauses $(u \vee v)$ in $\mathcal{F} - X$. We create an instance of DIGRAPH PAIR CUT(k) according to the following procedure.

Reduction. Intuitively, vertices in the created graph correspond directly to variables of $\mathcal{F}$ and reachability of a vertex from s corresponds to setting the variable to 1; here zero-validity of $\mathcal{F} - X$ is crucial for encoding clauses via arcs and pairs. For variables $x \in X$ we instead use two vertices x_0, x_1 of which one must always be deleted because they are neighbors of s and we add the pair $\{x_0, x_1\}$. If x_i is not deleted then this corresponds to setting x to i. The total budget is $k' = |X| + k$ consisting of budget $|X|$ for deletion of one of x_0 and x_1 for each $x \in X$ and budget k for the actual deletions: Deleting both x_0 and x_1 instead of one costs 1 extra and corresponds to deletion of variable x; for variables $v \in V \setminus X$ deletion of vertex v directly corresponds to deletion of v in $\mathcal{F}$. Formally, we proceed as follows:

1. Create a directed graph D with vertex set $V' = \{s\} \cup \{x_0, x_1 \mid x \in X\} \cup (V \setminus X)$ and with an arc (s, x_i) for every $x \in X$, $i \in \{0, 1\}$.

2. Create an initial set of pairs $\mathcal{P} = \{\{x_0, x_1\} \mid x \in X\}$.

3. For clauses on two variables $x, y \in X$ add the pair $\{x_i, y_j\}$ to $\mathcal{P}$ where $x = i$ and $y = j$ is the unique assignment that falsifies the clause.

4. For clauses on two variables $u, v \in V \setminus X$, taking into account symmetry and zero-validity of $\mathcal{F} - X$, there are only two cases:

   (a) For clauses $(\neg u \vee v) \equiv (u \rightarrow v)$ add the arc (u, v) to D. If neither vertex is deleted then this prevents that u is reachable but v is not, which corresponds to the unique assignment $u = 1$ and $v = 0$ that falsifies the clause.

   (b) For clauses $(\neg u \vee \neg v)$ add the pair $\{u, v\}$ to D. If neither vertex is deleted then this prevents that both are reachable, which corresponds to the unique assignment $u = 1$ and $v = 1$ that falsifies the clause.

5. For clauses on two variables $x \in X$ and $v \in V \setminus X$ there are four cases, including also the positive clause $(x \vee v)$. Because we can use x_0 and x_1, in particular x_0 to express $\neg x$, they can be treated in the same way as the previous two:

   (a) For clauses $(x \vee v) \equiv (\neg x \rightarrow v)$ add the arc (x_0, v) to D.

   (b) For clauses $(\neg x \vee v) \equiv (x \rightarrow v)$ add the arc (x_1, v) to D.

   (c) For clauses $(x \vee \neg v) \equiv (\neg(\neg x) \vee \neg v)$ add the pair $\{x_0, v\}$ to $\mathcal{P}$.

   (d) For clauses $(\neg x \vee \neg v)$ add the pair $\{x_1, v\}$ to $\mathcal{P}$.

6. Output the instance with directed graph $D = (V', A)$, source vertex s, pairs $\mathcal{P}$, and budget $k' = k + |X|$.

We note that if $|X| \geq k$, which can readily be assumed since otherwise the instance is trivially **yes**, then this a polynomial parameter transformation from ALMOST 2-SAT COMPRESSION($|X|$) to DIGRAPH PAIR CUT(k). See Lemma 17 below for a sketch of a proof. Combined with an approximate solution X of size $k^{\mathcal{O}(1)}$, this can be used for a polynomial kernelization of ALMOST 2-SAT(k) (as was done in the preliminary version of this article [46]). However, doing so directly increases the problem parameter from k to $k + |X|$, which both increases the running time of an FPT algorithm and prevents us from usefully running an approximation algorithm on the output. Instead we show how to compute a solution-covering set for $\mathcal{F}$ directly.

Lemma 17. *There is a randomized polynomial-time algorithm that takes an input $(\mathcal{F}, k, X)$ of* ALMOST 2-SAT COMPRESSION$(|X|)$ *where $k < |X|$ and computes a set $Y \supseteq X$ of $\mathcal{O}(|X|^4)$ variables such that Y contains a minimum-cardinality deletion set for $\mathcal{F}$. The failure probability is exponentially small in the input size.*

Proof. Let an instance $(\mathcal{F}, k, X)$ of ALMOST 2-SAT COMPRESSION$(|X|)$ be given and assume without loss of generality that $\mathcal{F} - X$ is zero-valid. Reduce $(\mathcal{F}, k, X)$ to an instance $(D, s, \mathcal{P}, k)$ of DIGRAPH PAIR CUT(k) using the reduction given above. Let V be the set of variables of $\mathcal{F}$, and let $X' = \{x_0, x_1 \mid x \in X\}$. Use Lemma 15 with input $(D, s, \mathcal{P}, |X|)$ to reduce $\mathcal{P}$ to a representative set of $\mathcal{O}(|X|^2)$ pairs; let $W = X' \cup \bigcup \mathcal{P}$ be the vertices contained in the remaining pairs together with all vertices in X'. Use Theorem 1 to compute a cut-covering set Z for 2-way cuts between subsets of X' and W in D, and define $Y = X \cup (Z \setminus X')$ and $Y' = W \cup Z$. We have $|W| = \mathcal{O}(|X|^2)$, and $|Y| = \mathcal{O}(|X||W||X|) = \mathcal{O}(|X|^4)$. Clearly, all steps can be performed in randomized polynomial time. We show that Y contains a minimum-cardinality deletion set for $\mathcal{F}$.

To this end, let B be a minimum-cardinality deletion set for $\mathcal{F}$, where $|B| \leq |X|$, and let $\phi : (V \setminus B) \to \{0, 1\}$ be a satisfying assignment for $\mathcal{F} - B$. Define

$$B' = \{x_0, x_1 \mid x \in B \cap X\} \cup \{x_{1-i} \mid \phi(x) = i, x \in X \setminus B\} \cup (B \setminus X),$$

i.e., B' coincides with B on $V \setminus X$ but it deletes both x_0 and x_1 when B deletes $x \in X$ and otherwise deletes x_i *not* matching the assignment $\phi(x)$. Clearly, $|B'| = |B| + |X|$.

We claim that for every vertex $v \in V \setminus X$ which is reachable from s in $D - B'$ we have $\phi(v) = 1$. By construction, for every $v \in V \setminus X$, there is an arc (α, v) in D exactly when there is a 2-clause in $\mathcal{F}$ enforcing $(\phi(\alpha) = 1 \to \phi(v) = 1)$. For every $x \in X$, the vertex x_i remains in $D - B'$ exactly if $x \notin B$ and $\phi(x) = i$, and due to the arc (s, x_i) it is then always reachable. Hence the claim follows.

Next define $T = (B' \cap W) \cup \{v \in W \setminus X' \mid \phi(v) = 0\}$, i.e., T contains vertices from W which are either deleted by B' (and recall $X' \subseteq W$) or which are set to false by ϕ. By the previous paragraph, B' separates T from s in D. It can also be seen that T is a $\mathcal{P}$-transversal, although we will not explicitly need this fact in this proof.

Now, since Y' is a cut-covering set, there is a minimum (X', T)-vertex cut $\hat{B}' \subseteq Y'$. Consider such a vertex cut, and define $\hat{B} \subseteq Y$ as $\{x \mid x_0, x_1 \in \hat{B}'\} \cup (\hat{B}' \setminus X')$. We will show that $\hat{B}$ is a deletion set for $\mathcal{F}$, and that $|\hat{B}| \leq |B|$. For the former, let $R \subseteq V(D)$ be the set of vertices reachable from $X' \setminus \hat{B}'$ in $D - \hat{B}'$. Define an assignment $\phi^* : (V \setminus \hat{B}) \to \{0, 1\}$ as

$$\phi^*(v) = \begin{cases} i & v \in X, v_{1-i} \in \hat{B}', v_i \notin \hat{B}' \\ 1 & v \in V \setminus X, v \in R \\ 0 & v \in V \setminus X, v \notin R. \end{cases}$$

Note that this exhausts the cases for variables in $V \setminus \hat{B}$, i.e., $\phi^*(v)$ is defined if and only if $v \notin \hat{B}$. We argue that ϕ^* is a satisfying assignment for $\mathcal{F} - \hat{B}$.

For this, assume to the contrary that there is a 2-clause $C \in \mathcal{F}$ that does not intersect $\hat{B}$ and which is falsified by ϕ^*, i.e., ϕ^* assigns values to both variables of C such that C is not satisfied by the assignment. Consider the cases.

- First assume that C intersects X. By construction, for every $x \in X$, if $x \in B$ then $x \in \hat{B}$, and if $\phi^*(x) = i$ then $\phi(x) = i$. Indeed, since $B' \cap X' \subseteq X' \cap T$, every (X', T)-cut, including $\hat{B}'$, must contain $B' \cap X'$. So if $x \in B \cap X$ then $x_0, x_1 \in X' \cap B'$, hence $x_0, x_1 \in \hat{B}'$ and $x \in \hat{B}$. Similarly, if $\phi^*(x) = i$ then $x_i \notin \hat{B}'$, hence $x_i \notin B'$ and $\phi(x) = i$. Hence no clause with both variables contained in X is falsified by ϕ^*, and C must contain two variables x, v where $x \in X$ and $v \in (V \setminus X)$, where furthermore $\phi(x) = \phi^*(x)$ and this assignment fails to satisfy C. Let $\phi(x) = i$, $i \in \{0, 1\}$ so that $x_i \notin B' \cup \hat{B}'$. There are two cases. If v occurs negatively in C, then there is a pair $\{x_i, v\}$ in $\mathcal{P}$, and we must have $v \in T$. But then $v \notin R$ by definition of $\hat{B}'$, which contradicts the assumption that $\phi^*(v) = 1$. If v occurs positively in C, then there is an arc (x_i, v) in D and since $x_i \notin \hat{B}'$ we conclude $v \in R$ and $\phi^*(v) = 1$, satisfying C.

- Otherwise C is disjoint from X, and by assumption C is zero-valid. If $C = (\neg u \vee \neg v)$ for some $u, v \in V \setminus X$, then there is a pair $\{u, v\}$ in $\mathcal{P}$. Since ϕ satisfies C, assume w.l.o.g. that $\phi(u) = 0$. But then $u \in T$ and $\hat{B}'$ separates u from s, implying $u \notin R$. This contradicts $\phi^*(u) = \phi^*(v) = 1$.

- In the remaining case, we have w.l.o.g. $C = (\neg u \vee v)$, $u, v \in (V \setminus X)$, and there is an arc (u, v) in D. Since ϕ^* falsifies C, we have $\phi^*(u) = 1$ and $\phi^*(v) = 0$, implying $u \in R$ and $v \in V \setminus (R \cup \hat{B})$. But this contradicts the existence of the arc (u, v).

We conclude that every 2-clause in C is either satisfied by ϕ^* or intersects $\hat{B}$, i.e., $\hat{B}$ is a deletion set for $\mathcal{F}$. Furthermore $|\hat{B}| \leq |B|$: Note that

$$|\hat{B}| = |\{x \in X \mid x_0, x_1 \in \hat{B}'\}| + |\hat{B}' \setminus X'| = |\hat{B}'| - |X|,$$

and the same equation holds for B and B'. By choice of $\hat{B}'$, we have $|\hat{B}'| \leq |B'|$; hence $|\hat{B}| \leq |B|$. Thus $\mathcal{F}$ has an optimal deletion set contained in $Y = X \cup (Z \setminus X')$. ☐

We next show how to modify the torso operation to apply to 2-CNF formulas.

Lemma 18. *Let $\mathcal{F}$ be a 2-CNF formula on variable set V, and let $Y \subseteq V$. There is a polynomial-time procedure that computes a 2-CNF formula $\mathcal{F}'$, possibly including 1-clauses, on variable set Y such that for every $X \subseteq Y$, X is a deletion set for $\mathcal{F}'$ if and only if X is a deletion set for $\mathcal{F}$.*

Proof. The procedure is analogous to the torso operation on graphs. Let $U := V \setminus Y$, and for a set $S \subseteq V$, define $\mathcal{F}[S]$ as the set of clauses of $\mathcal{F}$ whose variables are contained in S. Begin with $\mathcal{F}' := \mathcal{F}[Y]$. For all $y \in Y$ and all $\phi' \colon \{y\} \to \{0, 1\}$, if there is no satisfying assignment for $\mathcal{F}[U \cup \{y\}]$ with $y = \phi'(y)$ then add a 1-clause equivalent to $(y \neq \phi'(y))$. Similarly, for all $y, y' \in Y$ and all $\phi' \colon \{y, y'\} \to \{0, 1\}$, if there is no satisfying assignment for $\mathcal{F}[U \cup \{y, y'\}]$ with $y = \phi'(y)$ and $y' = \phi'(y')$ then add a 2-clause equivalent to $(y \neq \phi'(y) \vee y' \neq \phi'(y'))$. We claim that the resulting formula $\mathcal{F}'$ has the desired properties.

Let $X \subseteq Y$, and assume first that X is a deletion set for $\mathcal{F}'$. Let $\phi' \colon Y \setminus X \to \{0, 1\}$ be a satisfying assignment for $\mathcal{F}' - X$. We claim that X is also a deletion set for $\mathcal{F}$. To this end, we extend ϕ' to a satisfying assignment for $\mathcal{F} - X$ as follows: For $y \in Y \setminus X$ and $c \in \{0, 1\}$, define $U_c(y) \subseteq U$ as the set of variables in U whose value is forced to c by setting y to $\phi'(y)$ considering only clauses of $\mathcal{F}[U \cup \{y\}]$. That is, begin with adding to $U_c(y)$ all variables u that have a 2-clause equivalent to $(y = \phi'(y) \to u = c)$. Furthermore, apply the following rule exhaustively: If $u \in U_c(y)$ and there is a 2-clause equivalent to $(u = c \to u' = c')$ then add u' to $U_{c'}(y)$.

Observe that when $U_0(y) \cap U_1(y) \neq \emptyset$ then the formula $\mathcal{F}[U \cup \{y\}]$ is not satisfiable when $y = \phi'(y)$. The same is true if there is a 1-clause equivalent to $(u \neq c)$ but $u \in U_c(y)$. In both cases, by construction of $\mathcal{F}'$, we must have added a 1-clause equivalent to $(y \neq \phi'(y))$. This, however, contradicts the fact that ϕ' is a satisfying assignment for $\mathcal{F}' - X$. Similarly, if there are variables $y, y' \in Y \setminus X$ with $U_0(y) \cap U_1(y') \neq \emptyset$ or $U_1(y) \cap U_0(y') \neq \emptyset$ then $\mathcal{F}[U \cup \{y, y'\}]$ is not satisfiable for $y = \phi'(y)$ and $y' = \phi'(y')$. By construction of $\mathcal{F}'$ we must have added a clause preventing this assignment, which again contradicts the choice of ϕ'. It follows that all pairs of sets $U_0(y)$ and $U_1(y')$ are disjoint for all $y, y' \in Y \setminus X$ and do not contradict 1-clauses on variables in U. Thus, ϕ' can be extended with assignments $\phi(u) = c$ for every $u \in U_c(y)$, $y \in Y$, $c \in \{0, 1\}$. Let V' be the scope of this extended assignment ϕ. Now note that the 2-clauses of $\mathcal{F} - X$ not satisfied by ϕ are exactly those of $\mathcal{F} - (V' \cup X)$, as any other case would imply a further propagation in the construction of the sets $U_c(y)$ above. Furthermore, $\mathcal{F} - (V' \cup X) \subseteq \mathcal{F}[U]$ and hence it is satisfiable by assumption. Therefore we can extend ϕ to a satisfying assignment for $\mathcal{F} - X$ and X is a deletion set for $\mathcal{F}$.

Now assume that $X \subseteq Y$ is a deletion set of $\mathcal{F}$. Let $\phi \colon V \setminus X \to \{0, 1\}$ be a satisfying assignment for $\mathcal{F} - X$. Let ϕ' denote the restriction of ϕ to Y. Clearly, ϕ' is a satisfying assignment for $\mathcal{F}[Y \setminus X]$. It suffices to show that ϕ' also satisfies all clauses that were added when constructing $\mathcal{F}'$. Assume, for contradiction, that some added 2-clause on variables $y, y' \in Y \setminus X$ is not satisfied, say $(y \neq c \vee y' \neq c')$ for $c, c' \in \{0, 1\}$. By construction, there is no satisfying assignment for $\mathcal{F}[U \cup \{y, y'\}]$ that extends $y = c$ and $y' = c'$. This, however, contradicts the fact that ϕ is a satisfying assignment for $(\mathcal{F} - X) \supseteq \mathcal{F}[U \cup \{y, y'\}]$. The argument for a falsified 1-clause is analogous. Hence X is a deletion set for $\mathcal{F}'$. ☐

We can now show the kernel for ALMOST 2-SAT COMPRESSION($|X|$).

Theorem 16. ALMOST 2-SAT COMPRESSION($|X|$) *has a randomized polynomial kernel with $\mathcal{O}(|X|^4)$ variables and false negatives only. The failure probability is exponentially small in the input size.*

Proof. Let $(\mathcal{F}, k, X)$ be an instance of ALMOST 2-SAT COMPRESSION($|X|$). If $k \geq |X|$ return a dummy **yes**-instance, otherwise use Lemma 17 to compute a set $Y \supseteq X$ of $\mathcal{O}(|X|^4)$ variables such that Y contains a minimum-cardinality deletion set for $\mathcal{F}$. Use Lemma 18 to compute a 2-CNF formula $\mathcal{F}'$ on variable set Y. Finally, replace 1-clauses (y) or $(\neg y)$ in $\mathcal{F}'$ by 2-clauses using new variables, as described at the start of the section, ignoring duplicates. This introduces at most $2|Y|$ new variables. The kernel for ALMOST 2-SAT COMPRESSION($|X|$) is now simply the instance $(\mathcal{F}', k, X)$.

The running time and failure probability of the process follows from the respective lemmas, and the one-sided error follows from the torso-like operation of Lemma 18. In particular, note that all operations performed in Lemma 17, such as the reduction of the number of pairs $\mathcal{P}$ via the representative sets lemma, is only used to produce the variable set Y; hence the only possible consequence of bad randomness is that Y fails to contain an optimal deletion set.

To show correctness, first assume that $(\mathcal{F}, k, X)$ is **yes**, and let X' be a deletion set for $\mathcal{F}$, $|X'| \leq k$. There is then a deletion set $X^* \subseteq Y$ with $|X^*| \leq |X'|$ by Lemma 17, and by Lemma 18, X^* is also a deletion set for $\mathcal{F}'$. Hence $(\mathcal{F}', k, X)$ is **yes**. On the other hand, assume that $(\mathcal{F}', k, X)$ is **yes**, with a deletion set X^*. Then by Lemma 18 X^* is also a deletion set for $\mathcal{F}$ and $(\mathcal{F}, k, X)$ is **yes**. Therefore the instances are equivalent. Finally, we note that X is a deletion set for $\mathcal{F}'$ since X is a deletion set for $\mathcal{F}$ and $X \subseteq Y$, hence $(\mathcal{F}', k, X)$ is a valid instance of ALMOST 2-SAT COMPRESSION($|X|$). ☐

Theorem 17. ALMOST 2-SAT *admits a randomized polynomial-time $\mathcal{O}(\sqrt{\log OPT})$-approximation.*

Proof. For brevity, let $k_0 = \text{OPT}$. We show the result in steps, by first producing a solution of size $\mathcal{O}(k_0^{1.5})$, then running Theorem 16 with this solution to produce an output of $\mathcal{O}(k_0^6)$ vertices, then finally running an approximation algorithm on this output for an approximation ratio of $\mathcal{O}(\sqrt{\log k_0^6}) = \mathcal{O}(\sqrt{\log k_0})$.

We first run an FPT algorithm for ALMOST 2-SAT(k) with parameter k ranging from 1 to $\lfloor \log n \rfloor$. This can be done in polynomial time by using, e.g., the $\mathcal{O}^*(2.3146^k)$-time algorithm of Lokshtanov et al. [50]. If we are successful, we output an optimal solution in polynomial time, otherwise we have concluded that $k_0 \geq \log n$. Next we run the $\mathcal{O}(\sqrt{\log n})$-approximation of Agarwal et al. [3] (using that we can freely convert between clause- and variable-deletion variant), producing a solution X of size $|X| = \mathcal{O}(k_0\sqrt{\log n}) = \mathcal{O}(k_0\sqrt{k_0})$. Finally, for every value of k from $\lceil \log n \rceil$ to $|X| - 1$, we apply Theorem 16 to the instance $(\mathcal{F}, k, X)$ and run the algorithm of Agarwal et al. [3] on the output, producing a new solution X' for every value of k. We verify for each such set X' whether it is a deletion set for the original instance $\mathcal{F}$, and output the smallest deletion set verified this way. Clearly, the procedure runs in randomized polynomial time.

We argue correctness. If $k_0 < \log n$, then correctness is clear, hence assume $k_0 \geq \log n$ and $|X| = \mathcal{O}(k_0^{1.5})$. Now consider the execution of Theorem 16 with parameter value $k = k_0$. Then $(\mathcal{F}, k_0, X)$ is **yes** and the kernelization produces a kernel $(\mathcal{F}', k_0, X)$ with $\mathcal{O}(|X|^4) = \mathcal{O}(k_0^6)$ variables. By the correctness of the kernelization, $\mathcal{F}'$ has a deletion set of size k_0, and the approximation algorithm returns a solution X' of size $\mathcal{O}(k_0\sqrt{\log k_0^6}) = \mathcal{O}(k_0\sqrt{\log k_0})$. Let $Y \subseteq V(\mathcal{F})$ be the set that was produced by Lemma 17 in the kernelization. We may assume that $X' \subseteq Y$, since any additional variables of $\mathcal{F}'$ were simply introduced to implement 1-clauses (y) or $(\neg y)$, and can be replaced in X' by the original variable y. Then by Lemma 18, X' is also a deletion set for $\mathcal{F}$. By the verification procedure, any output we produce must be a deletion set for $\mathcal{F}$ of size at most $|X'|$, and we are done. ☐

Theorem 18. ALMOST 2-SAT(k) *admits a randomized kernelization with $\mathcal{O}(k^4 \log^2 k)$ variables, parameter value $k' \leq k$, with errors limited to false negatives and failure probability exponentially small in the input size.*

Proof. Let $(\mathcal{F}, k)$ be the input instance. Use Theorem 17 to produce a deletion set X for $\mathcal{F}$. Let k_0 be the size of an optimal solution. Then there are constants c and d such that $|X| \leq c(k_0\sqrt{\log k_0}) + d$. If $|X| > c(k\sqrt{\log k}) + d$, we reject the instance by producing a dummy **no**-instance, otherwise we run

Theorem 16 on $(\mathcal{F}, k, X)$ to compute an instance $(\mathcal{F}', k, X)$, and we produce the output $(\mathcal{F}', k)$ as kernel, where $\mathcal{F}'$ has $\mathcal{O}(|X|^4) = \mathcal{O}(k^4 \log^2 k)$ vertices. $\square$

6.3 Implications for related problems

In this section, we show some consequences of the ALMOST 2-SAT kernel. As ALMOST 2-SAT is quite a general problem, several interesting problems have been shown to reduce to it via parameter-preserving reductions (PPTs). Many of these reductions are surveyed by Raman et al. [66]. For the remaining problem, RHORN-BACKDOOR DELETION SET, we need a few definitions. Let $\mathcal{F}$ be a CNF formula on a variable set V. To *flip* a set $S \subseteq V$ of variables in $\mathcal{F}$ refers to negating all occurrences in $\mathcal{F}$ of variables $v \in S$, i.e., for every $v \in S$ we replace v by $\neg v$ and $\neg v$ by v. A CNF formula $\mathcal{F}$ is *Horn* if every clause contains at most one positive literal, and *renamable Horn* (RHorn) if one can flip a set of variables in $\mathcal{F}$ so that the result is Horn. SAT for Horn or RHorn formulas can be solved in polynomial time [33]. The problem RHORN-BACKDOOR DELETION SET, on input $(\mathcal{F}, k)$, refers to the problem of finding a set of at most k variables X in $\mathcal{F}$ such that deleting all occurrences of variables in X from clauses of $\mathcal{F}$ leaves a RHorn formula [20, 33]. RHORN-BACKDOOR DELETION SET is known to reduce to VERTEX COVER ABOVE MATCHING [33].

All in all, we get the following corollary.

Corollary 6. *The following problems have randomized polynomial kernels:* VERTEX COVER ABOVE MATCHING, VERTEX COVER PARAMETERIZED BY KŐNIG DELETION SET, KŐNIG VERTEX DELETION *restricted to input graphs having perfect matchings, and* RHORN-BACKDOOR DELETION SET.

Proof. All listed problems are NP-hard even with parameter value encoded in unary, since the maximum cost of any solution is already polynomially bounded. Therefore, it suffices to refer to the appropriate PPTs (see Section 2): VERTEX COVER ABOVE MATCHING and ALMOST 2-SAT(k) are equivalent under PPTs [66]. The following problems are reducible to VERTEX COVER ABOVE MATCHING under PPTs: KŐNIG VERTEX DELETION restricted to input graphs having perfect matchings [58] and RHORN-BACKDOOR DELETION SET [33].

For VERTEX COVER PARAMETERIZED BY KŐNIG DELETION SET, i.e., given X such that $G - X$ is a Kőnig graph, parameterized by $|X|$, one may observe the following: If the target value, say ℓ, is larger than a minimum vertex cover of $G - X$ (equal its maximum matching size, say $m(G - X)$) plus $|X|$ then the answer is trivially **yes**. Otherwise, from $\ell < |X| + m(G - X)$ it follows that $\ell - m(G) \leq \ell - m(G - X) < |X|$. Hence, by accounting for the target value as $k := \ell - m(G)$, i.e., "above maximum matching", the parameter can only decrease and we have an immediate PPT. $\square$

The result for VERTEX COVER PARAMETERIZED BY KŐNIG DELETION SET generalizes the best known kernelization result for VERTEX COVER PARAMETERIZED BY A FEEDBACK VERTEX SET [42]. There the input contains X such that $G - X$ is a forest, parameterized by $|X|$; as forests are also Kőnig, the parameter can only decrease by reducing to parameterization by Kőnig deletion set. The same is true in comparison to parameterization by an odd cycle transversal X.

We also provide a new reduction result in the form of a PPT from VERTEX COVER ABOVE LP to VERTEX COVER ABOVE MAXIMUM MATCHING. This implies that the two problems are equivalent; the reverse direction holds trivially since the parameter can only decrease when comparing the target value to the LP cost instead of to a maximum matching.

Lemma 19. *There is a polynomial parameter transformation from* VERTEX COVER ABOVE LP *to* VERTEX COVER ABOVE MAXIMUM MATCHING.

Proof. Let (G, k) be an instance of VERTEX COVER ABOVE LP, i.e., asking whether G has a vertex cover of size at most $LP(G) + k$, where $LP(G)$ denotes the minimum cost of a fractional vertex cover for G. Let $\ell = LP(G) + k$. It is well known that the linear programming relaxation for VERTEX COVER is half-integral [5, 62], so we will only consider such solutions.

It suffices for us to show that G contains a large matching. For the argument let us fix any minimum integral vertex cover X of G, and assume that the instance is **yes**, so $|X| \leq \ell$. Create a graph G' by adding a set of $2k + 1$ vertices Y, and making X and Y fully adjacent. Let us consider optimal fractional vertex covers of G'. We may clearly assume that each vertex of Y is assigned the same value in such a cover. We also note that X is a vertex cover of G', hence $LP(G') \leq |X|$.

If all vertices of Y are assigned $\frac{1}{2}$, or all are assigned 1, then the total cost is at least

$$LP(G) + \tfrac{1}{2}|Y| > LP(G) + k \geq \ell \geq |X|,$$

using the simple fact that G' always incurs a cost of at least $LP(G)$ on the induced subgraph G in G'. Thus, optimal fractional covers of G' will assign zero to all of Y, since otherwise a cost exceeding X is incurred. Setting Y to zero already forces setting X to one, giving the unique optimal fractional solution (with all further vertices also set to zero).

Thus the graph G' has a vertex cover X of size equal to the minimum fractional cost $LP(G')$. It is easy to see that it also has a matching of size $|X| = LP(G')$: Assume that the vertices of X cannot be matched into $V(G') \setminus X$. It follows, by Hall's Theorem, that there is a subset X' of X with $|N_{G'}(X')| < |X'|$. But then assigning $\frac{1}{2}$ to X' and $N(X')$ would give a cheaper fractional solution, a contradiction.

It follows that G' has a matching of size $|X|$, implying that $G = G' - Y$ has a matching M of size at least $|X| - |Y| \geq LP(G) - (2k+1)$. Thus we get that our target value ℓ exceeds the size of a maximum matching by at most $\ell - |M| \leq \ell - (LP(G) - (2k+1)) = \ell - LP(G) + 2k + 1 \leq 3k + 1$.

Note that we do not need to know whether (G, k) is **yes**, nor do we need to know X. We may compute a maximum matching M of G and check whether it meets at least the bound which we proved for the **yes**-case. If so, then (G, k') with $k' = \ell - |M| = LP(G) + k - |M|$ is an equivalent instance of VERTEX COVER ABOVE MAXIMUM MATCHING with $k' \leq 3k + 1$. Otherwise, we know that (G, k, ℓ) is **no**, and we return a dummy **no**-instance. □

As an immediate corollary we get that VERTEX COVER ABOVE LP admits a polynomial kernel.

Corollary 7. VERTEX COVER ABOVE LP *admits a randomized polynomial kernelization.*

7 Multiway cut and feedback problems

In this section we use the s-way cut-covering sets of Theorem 2 to obtain polynomial kernelizations for s-MULTIWAY CUT(k), s-MULTICUT(k), and Γ-FEEDBACK VERTEX SET(k). The following theorem summarizes the results of this section.

Theorem 19. *The following problems admit randomized polynomial kernelizations with error exponentially small in the input size.*

1. s-MULTIWAY CUT(k) *has a kernel with* $\mathcal{O}(k^{s+1})$ *vertices,*
2. s-MULTICUT(k) *has a kernel with* $\mathcal{O}(k^{\lceil\sqrt{2s}\rceil+1})$ *vertices, and*
3. Γ-FEEDBACK VERTEX SET(k) *has a kernel with* $\mathcal{O}((k \log k \log\log k)^{s+1})$ *vertices.*

The three parts of Theorem 19 are proven over the course of the following three subsections.

7.1 A polynomial kernelization for s-Multiway Cut

In this section we give a polynomial kernelization for s-MULTIWAY CUT(k), which is defined as follows.

s-MULTIWAY CUT(k) **Parameter:** k.
Input: An undirected graph $G = (V, E)$, a set $T \subseteq V$ of at most s terminals, and an integer $k \in \mathbb{N}$.
Question: Is there a set $X \subseteq V \setminus T$ such that each connected component of $G - X$ contains at most one terminal?

As a first step we briefly recall that using the LP-based approach of Guillemot [34] we can efficiently reduce any given instance to an equivalent one that has at most $2k$ vertices that are adjacent to terminals. This approach does not increase the number of terminals and, thus, we state it directly for s-MULTIWAY CUT(k).

Lemma 20 ([34]). *There is a polynomial-time algorithm that given any instance (G, T, k) of s-MULTIWAY CUT(k) returns an equivalent instance (G', T', k') with $|N_{G'}(T')| \leq 2k'$ and $k' \leq k$.*

For ease of presentation we will tacitly assume that given instances (G, T, k) already satisfy $|N(T)| \leq 2k$. We may also assume that no vertex is adjacent to more than one terminal since it is optimal (and mandatory) to delete such vertices to separate all terminals. Finally, we assume that no two terminals are adjacent since that would make the instance trivially negative.

The following lemma captures the kernelization result. The main argument is that using Theorem 2 to preserve an optimal s-way cut for every partition of $N(T)$ into at most s sets will also preserve an optimal solution for (G, T, k), if one exists.

Lemma 21. *s-MULTIWAY CUT(k) has a randomized polynomial kernelization that on input of (G, T, k) creates an equivalent instance (G', T, k) with $|V(G')| = \mathcal{O}(k^{s+1})$. The error probability is exponentially small in the input size.*

Proof. Given an instance (G, T, k), the kernelization proceeds by calling the algorithm of Theorem 2 on input graph G and terminal set $U := N(T)$, preserving optimal s-way cuts of U. (Note that the set T in the statement of Theorem 2 is distinct from the set T in the problem input (G, T, k).) In randomized polynomial time, with error probability exponentially small in the size of $|V(G)|$, this returns a set $Z \subseteq V$ of size $\mathcal{O}(k^{s+1})$ such that for every partition $\mathcal{U}$ of U the set Z contains a minimum multiway cut for $\mathcal{U}$ in G. The kernelization then applies the torso operation on $Z \cup U \cup T$ to obtain the graph G', i.e., by adding shortcut edges for all paths with internal vertices from $V \setminus (Z \cup U \cup T)$ and restricting to vertex set $Z \cup U \cup T$.

Assume that (G, T, k) is **yes**. Thus, there exists a multiway cut $X \subseteq V \setminus T$ such that in $G - X$ no component contains more than one terminal. It follows directly that all terminal neighbors of any $t \in T$ that are present in $G - X$, i.e., the set $N(t) \setminus X \subseteq U$, are in the same component. Thus, X is also a multiway cut of the partition $\mathcal{U}$ of U that partitions according to adjacency with T. (Recall that every vertex is adjacent with at most one terminal, and that no two terminals are adjacent.) Therefore, by Theorem 2, the set Z also contains a multiway cut X' for $\mathcal{U}$ of size at most $|X|$. Then X' is also a multiway cut for $\mathcal{U}$ in G', by the properties of the torso operation (Proposition 2). Furthermore, we may assume that $X' \cap T = \emptyset$, as $X' \setminus T$ would also be a multiway cut for $\mathcal{U}$. It follows that X' is a multiway cut for G', and (G', T, k) is **yes**.

For the converse, assume that (G', T, k) is **yes**, and let X be a multiway cut for G' with $|X| \leq k$. Then X still separates all terminals of T in G (by Proposition 2), hence X is a multiway cut for G and (G, T, k) is **yes**, too. ☐

7.2 Polynomial kernelization for s-Multicut

In this section we turn to the MULTICUT problem where, instead of separating every pair of terminals t_1, t_2 from a set T, we are given a family $\mathcal{T}$ of pairs and must separate every pair therein; in s-MULTICUT instances have at most s pairs to separate.

s-MULTICUT(k)	**Parameter:** k.
Input: An undirected graph $G = (V, E)$, a family $\mathcal{T} \subseteq \binom{V}{2}$ with $\lvert\mathcal{T}\rvert \leq s$, and $k \in \mathbb{N}$.	
Question: Is there a set $X \subseteq V \setminus V(\mathcal{T})$ such that in $G - X$ no component contains both vertices t_1, t_2 of any pair $\{t_1, t_2\} \in \mathcal{T}$?	

At high level we will proceed in a similar way as for the kernelization for s-MULTIWAY CUT(k). The key similarity is that a multicut X for $\mathcal{T}$ also splits the graph into components and thereby generates a partition of the terminals. However, one can show that there is always a partition into at most $\lceil\sqrt{2s}\rceil$ sets such that the requested multicut (if it exists) is also a multiway cut for that partition. Conversely, by preserving an optimal multiway cut for each partition into at most $\lceil\sqrt{2s}\rceil$ sets that has a cut of size at most k, we also preserve a multicut of size at most k for $\mathcal{T}$ (if one exists). As a minor difference, we work directly on the set of terminals instead of the terminal neighbors, making each terminal "undeleteable" by creating $k+1$ copies of it.

Lemma 22. *s-MULTICUT(k) has a randomized polynomial kernelization that on input of $(G, \mathcal{T}, k)$ creates an equivalent instance $(G', \mathcal{T}, k)$ with $|V(G')| = \mathcal{O}(k^{\lceil\sqrt{2s}\rceil+1})$. The error probability is exponentially small in the input size.*

Proof. Let $(G, \mathcal{T}, k)$ be an instance of s-MULTICUT(k) with $\mathcal{T} = \{\{a_1, b_1\}, \ldots, \{a_r, b_r\}\}$, for some $r \leq s$, and let $T = \{a_i, b_i \mid i \in [r]\}$. Replace every terminal $t \in T$ by a set of $k+1$ copies $t^{(1)}, \ldots, t^{(k+1)}$ that are pairwise adjacent and whose neighborhoods are identical to that of t, to simulate t being undeletable, and let $U = \{a_1^{(1)}, \ldots, b_r^{(k+1)}\}$ be the full set of resulting terminals. Note that U has size at most $2r(k+1) \leq 2s(k+1) = \mathcal{O}(k)$. Let G^+ denote the resulting graph. The kernelization proceeds by calling the algorithm of Theorem 2 on G^+ and terminal set U, for which optimal $\lceil\sqrt{2s}\rceil$-multiway cuts shall be preserved. In randomized polynomial time with error probability exponentially small in the size of $|V(G^+)| = \mathcal{O}(|V(G)|)$ this returns a set $Z^+ \subseteq V(G^+)$ of size $\mathcal{O}(k^{\lceil\sqrt{2s}\rceil+1})$ such that for every partition $\mathcal{U}$ of U into at most $\lceil\sqrt{2s}\rceil$ parts, the set Z^+ contains a minimum multiway cut for $\mathcal{U}$ in G^+. The kernelization then applies the torso operation on $Z \cup T$ to obtain the graph G' from G where $Z := Z^+ \cap V(G)$, i.e., Z contains only the non-terminal vertices in Z^+ but $Z \cup T$ contains those and all terminals in G.

Assume that $(G, \mathcal{T}, k)$ is **yes** and let $X \subseteq V \setminus T$ be a multicut of size at most k for $\mathcal{T}$ in G. Let $\mathcal{U}$ be a maximally coarse partition of U such that

1. any two $u, u' \in U$ that are in the same connected component of $G^+ - X$ are contained in the same set in the partition $\mathcal{U}$, and

2. for any two sets $U_1, U_2 \in \mathcal{U}$ there is a terminal pair $\{a_i, b_i\} \in \mathcal{T}$ such that U_1 contains copies of a_i and U_2 contains copies of b_i, or vice versa.

Such a partition $\mathcal{U}$ can be obtained as follows: First, because $X \subseteq V \setminus T$ and because copies $t^{(1)}, \ldots, t^{(k+1)}$ form a clique, all copies $t^{(i)}$ of each terminal $t \in T$ are present in $G^+ - X$ and they are in the same connected component. Now, start with the partition $\mathcal{U} = \mathcal{U}_0$ of U that puts two terminals $u, u' \in U$ in the same set if and only if they are in the same connected component of $G^+ - X$. Next, merge any two sets in $\mathcal{U}$ that do not fulfill (2); repeat this until all pairs of sets in $\mathcal{U}$ fulfill (2). Note that $\mathcal{U}_0$ has property (1) and that this is preserved by the merging operation. Conversely, if $\{t, t'\} \in \mathcal{T}$ then t and t' are in different connected components of $G - X$, and any two $t^{(i)}$ and $t'^{(j)}$ are in different components of $G^+ - X$. Thus no two $t^{(i)}$ and $t'^{(j)}$, for $\{t, t'\} \in \mathcal{T}$ are in the same set of $\mathcal{U}_0$, and nor are they in the same set of $\mathcal{U}$ as pair requests in $\mathcal{T}$ prevent merging of sets.

It is easy to see that condition (2) leads to an upper bound of $\lceil\sqrt{2s}\rceil$ for the number p of sets in $\mathcal{U}$ since every pair $U_1, U_2 \in \mathcal{U}$ have a different terminal pair for (2) as all copies $t^{(i)}$ of any terminal $t \in T$ are in the same component of $G^+ - X$. The bound can be derived as follows:

$$\begin{aligned} \binom{p}{2} \leq r \leq s &\implies p(p-1) \leq 2s \\ &\implies p \leq \frac{1}{2} + \sqrt{\frac{1}{4} + 2s} < \frac{1}{2} + \frac{1}{2} + \sqrt{2s} \\ &\implies p \leq \left\lceil \sqrt{2s} \right\rceil. \end{aligned}$$

Here, the final implication holds because p is integer.

Thus, for the partition $\mathcal{U}$ of U there is a multiway cut of size at most k in G^+, namely the set X. It follows that Z^+, as returned from Theorem 2 also contains a multiway cut X^+ for $\mathcal{U}$ of size at most $|X| \leq k$. Let $X' = X^+ \setminus U$, which also has size at most k and which has $X' \cap T = \emptyset$. By definition of Z and since $X^+ \subseteq Z^+$, it follows that $X' \subseteq V(G') = T \cup Z$. We show that X' is a multicut for $\mathcal{T}$ in G': Fix any $\{t, t'\} \in \mathcal{T}$, noting that $t, t' \in T$. By Proposition 3, there is a t, t'-path in $G - X'$ if and only if there is a t, t'-path in $G' - X'$, as the torso operation bypasses all vertices in $V \setminus (T \cup Z)$, $t, t' \in T$, and $X' \subseteq Z$. In other words, it suffices to show that X' separates t and t' in G. Accordingly, let $P = (t, v_1, \ldots, v_\ell, t')$ be any t, t'-path in G. Since $|X^+| \leq k$, at least one copy each of t and t' exists also in $G^+ - X^+$, say $t^{(i)}$ and $t'^{(j)}$. It follows directly that $P^+ = (t^{(i)}, v_1, \ldots, v_\ell, t'^{(j)})$ is a path in G^+ with $t^{(i)}, t'^{(j)} \notin X^+$. Moreover, as observed above, $t^{(i)}$ and $t'^{(j)}$ are in different sets of $\mathcal{U}$ and, thus, X^+ must contain one of the internal vertices of P^+, say $v_q \in V \setminus T$. Clearly, $v_q \in X' = X^+ \setminus U$, as required. Thus, X' is a solution for $(G', \mathcal{T}, k)$, implying that this instance is **yes**.

For the converse, recall that the torso operation will not decrease the cost of multicuts for $\mathcal{T}$ since it corresponds to making all vertices in $V \setminus (Z \cup T)$ undeletable, shrinking the space of feasible solutions. ☐

7.3 Polynomial kernelization for Γ-Feedback Vertex Set(k)

As a further consequence of Theorem 2 we show how to obtain a polynomial kernel for the following group cut problem. Let $(\Gamma, \otimes)$ be a finite group with unit element $1_\Gamma \in \Gamma$, and let $\Gamma = \{\alpha_1, \ldots, \alpha_s\}$ where $\alpha_1 = 1_\Gamma$.

Γ-FEEDBACK VERTEX SET(k)— Γ-FVS(k) **Parameter:** k.
Input: A directed graph $D = (V, A)$, an edge labeling $\phi : A \to \Gamma$, and an integer k.
Question: Is there a set X of at most k vertices such that $D - X$ has a consistent labeling, i.e., a function $\pi : V \setminus X \to \Gamma$ such that $\pi(u) \otimes \phi((u, v)) = \pi(v)$ for all arcs (u, v) of $D - X$.

We remark that the graph is directed only as a technical requirement, i.e., to define a direction for the element labels. In all practical aspects (e.g., when referring to cycles in D), we will treat the graph as an undirected graph (the underlying undirected graph of D), with edge labels that read differently in different directions. Thus, if $(u, v) \in A$ with label α, then we act as if in addition $(v, u) \in A$ with label α^{-1}. Since we are dealing with the vertex deletion variant, this does not cause any problems.

An equivalent definition is as follows [34]. Let a cycle C in the underlying undirected graph of D be called *non-null* (or non-unit) if its edge labels, read in the direction of traversal in the cycle, multiply to any value other than 1_Γ. Note that this is well-defined, and does not depend on the starting point or the direction of traversal of the cycle. The problem Γ-FVS(k) is then equivalent to asking for a set of at most k vertices which intersects all non-null cycles. This and similar problems were, e.g., studied by Guillemot [34] and Chudnovsky et al. [12]. Note that the well-known ODD CYCLE TRANSVERSAL problem is an instance of this problem, with the group Z_2. Guillemot [34] shows that GROUP FEEDBACK ARC SET(k) and GROUP FEEDBACK VERTEX SET($k + |\Gamma|$) (see below) are FPT, but leaves open the questions of polynomial kernels, and whether GROUP FEEDBACK VERTEX SET(k) is FPT. The latter question was answered affirmatively by Cygan et al. [16]; a faster algorithm was shown by Iwata et al. [41]. Our current results make partial progress towards the kernelization questions, but note that there is an easy reduction from MULTIWAY CUT(k) to GROUP FEEDBACK VERTEX SET(k): Use a group of size at least matching the number of terminals. Make a clique on the terminals and add correct edge labels to enforce a different value for each terminal; also make the terminals "heavy" by making $k + 1$ copies. All other edges have the unit label. It is easy to verify that group cuts in the resulting graph correspond directly to multiway cuts in the input graph.

As a first step towards our kernel we will require a polynomial-time procedure for producing an initial solution of size $k^{\mathcal{O}(1)}$. This we do by reducing our problem to a VERTEX MULTICUT setting, by splitting every vertex in D into s copies, in a way that can be seen as splitting vertices into "literals". Known approximation algorithms then provide us with the initial solution we need. We begin by showing that an approximation algorithm for VERTEX MULTICUT is implicit in the literature.

Lemma 23. VERTEX MULTICUT *has an* $\mathcal{O}(\log OPT \log\log OPT)$*-approximation, where* OPT *is the size of an optimal solution.*

Proof. The problem reduces to SYMMETRIC MULTICUT, which is defined as follows. The input is a directed graph $D = (V, A)$ with a capacity $c(e)$ on every arc $e \in A$, and a set of terminal pairs $S = \{(s_i, t_i) \mid i = 1, \ldots, p\}$. The task is to find a set $F \subseteq A$ of minimum total capacity such that no strong connected component of $D - F$ contains both members s_i, t_i of any terminal pair $(s_i, t_i) \in S$. The problem SYMMETRIC MULTICUT has an $\mathcal{O}(\log \mathrm{OPT} \log\log \mathrm{OPT})$-approximation, see Even et al. [24].

For the reduction, let $(G = (V, E), \mathcal{T})$ be an instance of VERTEX MULTICUT, $\mathcal{T} \subseteq \binom{V}{2}$. We create a digraph $D = (V', A)$ from G by replacing each edge $\{u, v\}$ in G by a pair of arcs (u, v), (v, u), then subdividing every vertex by a directed arc. Concretely, we let $V' = \{v^-, v^+ \mid v \in V\}$ and $A = \{(u^+, v^-) \mid \{u, v\} \in E\} \cup \{(v^-, v^+) \mid v \in V\}$. Furthermore, we let $S = \{(s_i^-, t_i^+) \mid \{s_i, t_i\} \in \mathcal{T}\}$ and assign capacity $c((v^-, v^+)) = 1$ for each $v \in V$ and $c(e) = |V| + 1$ for every other arc $e \in A$. For the correctness, observe that every optimal solution to (D, S) will only delete arcs (v^-, v^+), corresponding to deleting a vertex in G, since deleting all arcs (v^-, v^+) kills the path from s_i^- to t_i^+ for any $(s_i^-, t_i^+) \in S$. Thus consider a set of vertices $X \subseteq V$ and let $X' = \{(v^-, v^+) \mid v \in X\}$. We claim that X is a multicut of $(G, \mathcal{T})$ if and only if X' is a solution to (D, S). Indeed, on the one hand, if $G - X$ contains an (s, t)-path P for some pair $\{s, t\} \in \mathcal{T}$, then the copies in D of the vertices used in P contain both an (s^-, t^+)-path and an (t^+, s^-)-path, violating the pair $(s^-, t^+) \in S$. On the other hand, if $D - X'$ contains both an (s^-, t^+)-path and an (t^+, s^-)-path, then the copies in G of the vertices used in the (s_i^-, t_i^+)-path contain an (s_i, t_i)-path, violating the pair $\{s_i, t_i\} \in \mathcal{T}$. □

Lemma 24. *There is a polynomial-time algorithm that given an instance (D, ϕ, k) of Γ-FVS(k) either proves that it is negative or produces a solution of size $\mathcal{O}(|\Gamma| k \log(|\Gamma| k) \log\log(|\Gamma| k))$.*

Proof. Let $D = (V, A)$ and let $s := |\Gamma|$. We construct a graph $G = (V', E)$ with $V' = \{v(\alpha) \mid v \in V, \alpha \in \Gamma\}$ and $E = \{\{u(\alpha), v(\alpha \otimes \phi((u, v)))\} \mid (u, v) \in A, \alpha \in \Gamma\}$. These edges are chosen such that if $\{u(\alpha), v(\alpha')\} \in E$, and if π is a consistent labeling which labels both u and v, then $\pi(u) = \alpha$ if and only if $\pi(v) = \alpha'$. Finally, the set of terminal pairs is $T = \{(u(\alpha), u(\alpha')) \mid u \in V, \alpha, \alpha' \in \Gamma, \alpha \neq \alpha'\}$. We show that the optimal multicut of (G, T) is within a factor of s of the optimal group feedback vertex set of (D, ϕ).

Claim 2. *Let X be a group feedback vertex set for (D, ϕ) of size k. Then $X' = \{v(\alpha) \mid v \in X, \alpha \in \Gamma\}$ is a valid multicut of (G, T) of size sk.*

Proof of claim. Any path between terminal pairs in $G - X'$ would form a non-null cycle in $D - X$. □

Claim 3. *Let X be a multicut of (G, T). Let $V(X) = \bigcup_{v(\alpha) \in X} \{v\}$. Then $V(X)$ is a solution to the group cut problem of size at most $|X|$.*

Proof of claim. Consider a non-null cycle in $D - V(X)$, and a member v of this cycle. Arbitrarily select a literal $v(\alpha)$ of v, and let S be the set of literals propagated along the cycle, i.e., if u is neighbor to v in the cycle with an edge that prescribes $\pi(u) = \pi(v) \otimes \alpha'$, then add the literal $u(\alpha \otimes \alpha')$ to S. Do this for one revolution around the cycle, so that S contains two literals of v. The literals of S form a path in $G - X$ between different literals of v, which contradicts X being a multicut. □

Now, if (D, ϕ, k) is positive, then by Lemma 23 we can find a multicut X for (G, T) of size $\mathcal{O}(p \log p \log\log p)$ where $p \leq sk = |\Gamma| k$ is the size of a minimum multicut for (G, T). This multicut X can then be projected back into an initial feedback vertex set $V(X)$ for (D, ϕ) of at most the same size. Otherwise, if X is too big, we may reject (D, ϕ, k). □

For the remainder of the section let us fix one instance (D, ϕ, k) of Γ-FEEDBACK VERTEX SET(k) and let X be an approximate solution of size at most $\mathcal{O}(sk \log(sk) \log\log(sk))$ obtained via Lemma 24. A simple normalization is possible, in the form of an *untangling*. By a simple shifting of the arc labels, guided by the existence of a consistent labeling for $D - X$, we can modify the labeling ϕ such that every arc of $D - X$ gets the label 1_Γ, without changing the set of group feedback vertex sets. This will be very useful in treating Γ-FVS(k) as a cut problem.

Lemma 25. *There is a polynomial-time algorithm that takes an instance (D, ϕ, k) of Γ-FEEDBACK VERTEX SET(k) and a group feedback vertex set X for (D, ϕ), and produces a new labeling ϕ' of D such that the following hold:*

1. *For every arc (u, v) of $D - X$, we have $\phi((u, v)) = \phi((v, u)) = 1_\Gamma$.*

2. *Every cycle of D is non-null with respect to ϕ if and only if it is non-null with respect to ϕ'.*

Proof. Fix any consistent labeling $\pi\colon (V \setminus X) \to \Gamma$ for $D - X$. Intuitively, for every vertex $v \in V \setminus X$, we will shift the labels of arcs incident with v by the inverse of $\pi(v)$. This will have the desired effect. Let us begin by formally defining this shifting operation.

Claim 4. *Let $v \in V$ and $\alpha \in \Gamma$. Define* shifting *ϕ by α at v as the following operation:*

1. *For every arc (u, v) in D, let the new label of (u, v) be $\phi((u, v)) \otimes \alpha$*
2. *For every arc (v, u) in D, let the new label of (v, u) be $\alpha^{-1} \otimes \phi((v, u))$*

The shifting operation does not change the set of non-null cycles.

Proof of claim. Observe first that we maintain the required property $\phi((u, v)) = \phi((v, u))^{-1}$. Next, consider a cycle C in the underlying undirected graph of D. Assume that C passes through v in a sequence of vertices uvw. Then the product of the new arc labels of (u, v) and (v, w) is $\phi((u, v)) \otimes \alpha \otimes \alpha^{-1} \otimes \phi((v, w) = \phi((u, v)) \otimes \phi((v, w))$, i.e., unchanged by the shift. ☐

We modify ϕ by shifting v by $\pi(v)^{-1}$ for every $v \in V \setminus X$ in arbitrary order (note that this also modifies the labels of arcs incident with X). Let the new labeling be ϕ'. By the claim, the set of non-null cycles is unchanged by this operation. Furthermore, for any arc (u, v) in $D - X$ we have

$$\phi'((u, v)) = \pi(u) \otimes \phi((u, v)) \otimes \pi(v)^{-1} = 1_\Gamma,$$

where the last equality holds since by assumption, $\pi(u) \otimes \phi((u, v)) = \pi(v)$. ☐

Assume that the above modification has been applied to (D, ϕ) using X. As a side effect, note that the orientation of arcs of $D - X$ is now immaterial. Also assume that all arcs with exactly one endpoint in X are outgoing from X, and that X is an independent set in D. The latter can be arranged by subdividing arcs in $D[X]$, assigning new arc labels appropriately. Note that for any (possibly closed) path P in D with initial and final vertices in X and internal vertices in $V \setminus X$, the "resulting label" of P now depends only on its initial and final arcs. At this point, we could reduce the problem to MULTIWAY CUT with s terminals by guessing which vertices $x \in X$ are to be deleted, and guessing a label $\pi(x)$ to every non-deleted vertex x (details omitted).

To incorporate this idea into a kernelization, we instead proceed as follows. Create an undirected graph G by taking the underlying undirected graph of $D - X$, then for every vertex $x \in X$ and every label $\alpha \in \Gamma$ add a new vertex $x(\alpha)$ to G. Finally, connect $x(\alpha)$ to v if and only if x is incident with an arc (x, v) in D with $\phi((x, v)) = \alpha$. Let $T = \{(x, \alpha) \mid x \in X, \alpha \in \Gamma\}$. We will show that group feedback vertex sets for (D, ϕ) correspond to multiway cuts for certain generalized s-partitions of T in G, hence the kernelization can be completed by invoking Theorem 13.

We introduce some notation for convenience.

Definition 5. Let (G, T) and X be as above. For a set $S \subseteq T$, define $X(S) = \{x \in X \mid x(\alpha) \in S, \alpha \in \Gamma\}$, and for $S \subseteq X$ define $T(S) = \{x(\alpha) \mid x \in S, \alpha \in \Gamma\}$. A set $S \subseteq T$ is *regular* if $T(X(S)) = S$.

Lemma 26. *Let $(D = (V, A), \phi, k)$ be a Γ-FVS(k) instance and let $X \subseteq V$ be a group feedback vertex set for D such that every arc of $D - X$ has label 1_Γ, $D[X]$ is an independent set, and every arc incident with X is oriented out of X. Let G and T be defined from D and X as above. The instance (D, ϕ, k) is **yes** for Γ-FVS(k) if and only if there is a regular set $T_X \subseteq T$, a partition $\mathcal{T} = \{T_\alpha \mid \alpha \in \Gamma\}$ of $T \setminus T_X$ and a multiway cut Y of $\mathcal{T}$ in $G - T_X$ such that*

1. *if $x(\alpha) \in T_{\alpha'}$ and $x(\beta) \in T_{\beta'}$ then $\alpha' \otimes \alpha^{-1} = \beta' \otimes \beta^{-1}$, and*
2. *$|Y| + |X(T_X)| \leq k$.*

Moreover, if Y and $\mathcal{T}$ fulfill these properties then $X' = (Y \setminus T) \cup X(T_X \cup (Y \cap T))$ is a solution for (D, ϕ, k). We refer to a triple $(T_X, \mathcal{T}, Y)$ as a solution *for (G, T), and the* cost *of a solution is $|X(T_X \cup Y) \cup (Y \setminus T)|$.*

Proof. Assume first that (D, ϕ, k) is **yes** for Γ-FVS(k). Let $X' \subseteq V$ with $|X'| \leq k$ and let π be a consistent assignment for $D - X'$. Let $T_X = T(X' \cap X)$ and let $Y := X' \setminus X \subseteq V(G) \setminus T_X$. Note that $|Y| + |X(T_X)| \leq k$. Define a partition $\mathcal{T} = \{T_\alpha \mid \alpha \in \Gamma\}$ of $T \setminus T_X$ by adding each $x(\alpha) \in T \setminus T_X$ to the set $T_{\alpha'}$ with $\alpha' = \pi(x) \otimes \alpha$. Thus, if $x(\alpha) \in T_{\alpha'}$ and $x(\beta) \in T_{\beta'}$ then $\alpha' \otimes \alpha^{-1} = \pi(x) = \beta' \otimes \beta^{-1}$, as required. We claim that Y is a multiway cut for $\mathcal{T}$ in $G - T_X$.

For this, let P be a path in $G - (T_X \cup Y)$ with initial and final vertices in T and internal vertices in $V \setminus X$. Let $x(\alpha)$ and $\hat{x}(\hat{\alpha})$ be the initial and final vertices of P, respectively. Then there is a corresponding (possibly closed) path P' in $D - X'$, with initial and final vertices x and $\hat{x}$, and with initial and final arcs labeled α and $\hat{\alpha}$, respectively (the last arc traversed in reverse in P'). Furthermore, since π is a consistent assignment for $D - X'$, we have $\pi(x) \otimes \alpha \otimes \hat{\alpha}^{-1} = \pi(\hat{x})$, since every arc of P' except the initial and final have label 1_Γ. We thus have $\pi(x) \otimes \alpha = \pi(\hat{x}) \otimes \hat{\alpha}$, and by construction, $x(\alpha)$ and $\hat{x}(\hat{\alpha})$ belong to the same set in $\mathcal{T}$. Thus Y is a multiway cut for $\mathcal{T}$ in $G - T_X$, as claimed.

For the converse, assume that we have a set $T_X \subseteq T$, a partition $\mathcal{T} = \{T_\alpha \mid \alpha \in \Gamma\}$ of $T \setminus T_X$, and a set $Y \subseteq V(G) \setminus T_X$ fulfilling the properties stated in the lemma. We claim that (D, ϕ, k) is **yes** for Γ-FVS(k). Note that under the cost function we use, we may assume that T_X is a regular set, i.e., $T_X = T(S)$ for some $S \subseteq X$. Now define a set $X' = S \cup Y$. Clearly, $|X'| \leq k$; we claim that X' is a solution for (D, ϕ, k). To prove this, we construct a consistent assignment π for $D - X'$:

1. For $x \in X \setminus S$, let $\pi(x) = \alpha' \otimes \alpha^{-1}$ where $\alpha, \alpha' \in \Gamma$ such that $x(\alpha) \in T_{\alpha'}$. Note that $\pi(x)$ is well-defined by the first property assumed in the lemma statement.

2. For $v \in (V \setminus X)$ that are not contained in X' we define $\pi(v) = \alpha$ if there is an $\alpha \in \Gamma$ such that the component of v in $G - (T_X \cup Y)$ contains a vertex of T_α, or $\pi(v) = 1_\Gamma$ otherwise. Since Y is a multiway cut of $\mathcal{T}$ in $G - T_X$ this is well-defined.

To show consistency, consider an arc (u, v) in $D - X'$. There are two cases. If $u \in X$, then $v \in V \setminus X$ since X is independent. Assume $\phi((u, v) = \alpha$ and $\pi(v) = \alpha'$. Then $\pi(u) = \alpha' \otimes \alpha^{-1}$, hence $\pi(u) \otimes \alpha = \pi(v)$ and (u, v) is consistent. Otherwise, since all arcs are oriented out of X in D, we have $u, v \in V \setminus X$, hence $\phi((u, v)) = 1_\Gamma$. Then u and v are in the same connected component of $G - (T_X \cup Y)$, and $\pi(u) = \pi(v)$. In both cases, the arc (u, v) is consistent, showing that X' is a solution to (D, ϕ, k). This completes the proof. ☐

This allows us to construct a polynomial kernelization for the compression version of the problem. Formally, let Γ-FVS COMPRESSION denote the problem where the input is a Γ-FVS(k) instance $(D = (V, A), \phi, k)$ and additionally a group feedback vertex set $X \subseteq V$, and the parameter is $|X|$. We show a kernel for this problem.

Lemma 27. Γ-FVS COMPRESSION *has a randomized polynomial kernelization that on input* (D, ϕ, k, X) *creates an equivalent instance* (D', ϕ', k', X') *with* $|V(D')| = \mathcal{O}(|X|^{s+1})$. *The error probability is exponentially small in input size.*

Proof. Normalize (untangle) the instance using X as per Lemma 25, and create the undirected graph G and terminal set T from D and X as in Lemma 26. Let $Z \subseteq V(G)$ be a cut-covering set for generalized s-partitions of T in G, computed by Theorem 13. We will apply a torso operation for D down to $(Z \setminus T) \cup X$, modified to account for the labeling ϕ.

To this end, let G' be the result of applying the torso operation to G, down to vertex set $T \cup Z$. We argue that, subject to the computation of Z having succeeded, (G, T) admits a solution of cost at most k if and only if (G', T) does. Indeed, let $(T_X, \mathcal{T}, Y)$ be a solution for (G, T). By Theorem 13 we may assume $Y \subseteq Z$. Then both Y and T_X exist in G', and it is clear that for any pair of terminals $t, t' \in T \setminus T_X$, there is a path between t and t' in $G - (T_X \cup Y)$ if and only if there is such a path in $G' - (T_X \cup Y)$. Thus $(T_X, \mathcal{T}, Y)$ is a solution for (G', T). The reverse direction is argued similarly.

Next, we apply a cleaning step. Let $X_0 \subseteq X$ consist of those vertices $x \in X$ such that there exists a pair $x(\alpha), x(\beta) \in T$ connected by an edge in G', where $\alpha \neq \beta$. Furthermore, say that there is a *conflict* between $x \in X$ and $v \in Z \setminus T$ if v is adjacent to two distinct copies $x(\alpha), x(\beta)$ of x in T. Let $X_1 \subseteq X$

contain every vertex $x \in X$ that is involved in conflicts with more than k vertices v. Finally, delete the vertices $T(X_0 \cup X_1)$ from G', subdivide every remaining edge in $G'[T]$, and subdivide every remaining edge corresponding to a conflict. Let $X' = X \setminus (X_0 \cup X_1)$ and $k' = k - |X_0 \cup X_1|$. Let (G'', T') denote the result of these deletions and subdivisions. We claim that (G', T) admits a solution of cost at most k if and only if (G'', T') admits a solution with cost at most k'. The reverse direction is clear. For the forward direction, let $(T_X, \mathcal{T}, Y)$ be a solution for (G', T) of cost at most k with $Y \subseteq Z$, as argued above. For every $x \in X_0$, there is an edge between some $x(\alpha)$ and $x(\beta)$ in $G'[T]$; hence at least one of $x(\alpha), x(\beta) \in Y \cup T_X$ and $x \in X(T_X \cup Y)$, and we may view x as deleted. Similarly, for every conflict on a pair x and v, at least one of x and v is deleted, and if x is involved in more than k conflicts, then any solution of cost at most k must delete x. Hence we may assume that an optimal solution will delete every vertex in $X_0 \cup X_1$. Also, subdividing edges does not affect the solution status since there is always a minimum-cost solution that contains no subdividing vertex.

To wrap up, we define a directed graph D' on vertex set $X' \cup (V(G'') \setminus T')$ and an arc labeling ϕ' of D' to correspond to (G'', T') as follows: For each edge $\{u, v\}$ in G'' with $u, v \in V(G'') \setminus T'$ we arbitrarily add either (u, v) or (v, u) to D' and label it with 1_Γ. For each edge $\{x(\alpha), v\}$ in G'' with $x(\alpha) \in T'$ and $v \in V(G'') \setminus T'$ we add an arc (x, v) to D' and label it with α. Note that this exhausts all edges of G'' and that the result is a simple directed graph. We now observe that $V(D') \cap X = X'$, that X' is a group feedback vertex set for D', and that applying the reduction of Lemma 26 to (D', ϕ', k') and X' results in the graph G'' and terminal set T'. Hence, by Lemma 26 and by the sequence of claims made above, (D', ϕ', k', X') is **yes** for Γ-FVS COMPRESSION if and only if (D, ϕ, k, X) is. The running time and failure probability of the kernelization follows. Finally, the number of vertices of D' is $\mathcal{O}(|X|^{s+1} + |X|^2 + k|X|) = \mathcal{O}(|X|^{s+1})$, including the vertices introduced via subdivision. This finishes the proof. ☐

Lemma 28. Γ-FEEDBACK VERTEX SET(k) *for a group with s elements has a polynomial kernel with $\mathcal{O}((k \log k \log\log k)^{s+1})$ vertices and failure probability exponentially small in the input size, if s is a constant.*

Proof. The first result follows by combining Lemma 24 with Lemma 27 and cleaning up the dependency on $s = \mathcal{O}(1)$. Note that the output instance for Γ-FVS COMPRESSION can be reduced to an instance of Γ-FVS(k) by simply dropping the solution X. ☐

8 Conclusions

We give powerful new techniques for polynomial kernelization, centered around applications of a lemma from matroid theory due to Lovász [51] and Marx [53]. The resulting tools significantly advance the field of kernelization, and imply polynomial kernels for a range of problems, including ALMOST 2-SAT(k), s-MULTIWAY CUT(k), and s-MULTICUT(k) among other results. In particular, we show how the lemma can be applied to devise an *irrelevant vertex*-based approach for graph cut problems. In addition to the aforementioned kernels, this lets us find a form of cut-covering sets of small size: given a graph G and terminal set T, we can find a set Z, of size polynomial in $|T|$, such that for every $A, B \subseteq T$, a minimum (A, B)-vertex cut is contained in Z. Similarly, for a constant s, we can find a set Z of polynomial size such that for every partition of T (or a subset of T) into at most s blocks, a minimum multiway cut of the partition is contained in Z. We foresee further applications of these results. Similarly to Kratsch and Wahlström [47], our kernels are randomized; unlike in [47], they can all be made reduction rule-based. Furthermore, the failure probability can be reduced to be exponentially small in the input length, implying non-uniform polynomial-time kernelization.

Despite being randomized, we note that all our kernels are compatible with the lower bound framework of Bodlaender et al. [6] and Fortnow and Santhanam [31]; see [31, Corollary 3.4]. Hence, concrete polynomial upper and lower bounds for the current problems is a relevant path of research (see [19, 18, 39]). Similarly, it would be highly interesting to have deterministic kernelizations for the considered problems; likely this would mean to abandon the use of represented gammoids since deterministic, polynomial-time representations for gammoids are a longstanding open problem.

Further significant open questions include the existence of polynomial kernels for the general form of MULTIWAY CUT(k) and MULTICUT($s+k$), in edge- and vertex-deletion variants, and for the GROUP FEEDBACK ARC SET(k) and GROUP FEEDBACK VERTEX SET(k) problems with arbitrary groups. Additionally, a polynomial kernel for DIRECTED FEEDBACK VERTEX SET(k) remains an 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] Leonard M. Adleman. Two theorems on random polynomial time. In *19th Annual Symposium on Foundations of Computer Science, Ann Arbor, Michigan, USA, 16-18 October 1978*, pages 75–83. IEEE Computer Society, 1978.

[3] Amit Agarwal, Moses Charikar, Konstantin Makarychev, and Yury Makarychev. $O(\sqrt{\log n})$ approximation algorithms for Min UnCut, Min 2CNF deletion, and directed cut problems. In Harold N. Gabow and Ronald Fagin, editors, *STOC*, pages 573–581. ACM, 2005.

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

[5] M Balinski. On the maximum matching, minimum covering. In *Proc. Symp. Math. Programming*, pages 301–312. Princeton University Press, 1970.

[6] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.

[7] Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (meta) kernelization. *J. ACM*, 63(5):44:1–44:69, 2016.

[8] Hans L. Bodlaender, Fedor V. Fomin, and Saket Saurabh. Open problems, WORKER 2010, 2010. Available at `http://fpt.wikidot.com/open-problems`.

[9] Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Cross-composition: A new technique for kernelization lower bounds. In Thomas Schwentick and Christoph Dürr, editors, *STACS*, volume 9 of *LIPIcs*, pages 165–176. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011.

[10] Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo. Kernel bounds for disjoint cycles and disjoint paths. *Theor. Comput. Sci.*, 412(35):4570–4578, 2011.

[11] Rajesh Chitnis, Marek Cygan, MohammadTaghi Hajiaghayi, Marcin Pilipczuk, and Michal Pilipczuk. Designing FPT algorithms for cut problems using randomized contractions. *SIAM J. Comput.*, 45(4):1171–1229, 2016.

[12] Maria Chudnovsky, Jim Geelen, Bert Gerards, Luis A. Goddyn, Michael Lohman, and Paul D. Seymour. Packing non-zero a-paths in group-labelled graphs. *Combinatorica*, 26(5):521–532, 2006.

[13] Julia Chuzhoy. On vertex sparsifiers with steiner nodes. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 673–688. ACM, 2012.

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

[15] Marek Cygan, Stefan Kratsch, Marcin Pilipczuk, Michal Pilipczuk, and Magnus Wahlström. Clique cover and graph separation: New incompressibility results. *TOCT*, 6(2):6, 2014.

[16] Marek Cygan, Marcin Pilipczuk, and Michal Pilipczuk. On group feedback vertex set parameterized by the size of the cutset. *Algorithmica*, 74(2):630–642, 2016.

[17] Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. On multiway cut parameterized above lower bounds. *TOCT*, 5(1):3, 2013.

[18] Holger Dell and Dániel Marx. Kernelization of packing problems. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 68–81. SIAM, 2012.

[19] Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM*, 61(4):23, 2014.

[20] Bistra N. Dilkina, Carla P. Gomes, and Ashish Sabharwal. Tradeoffs in the complexity of backdoor detection. In Christian Bessiere, editor, *CP*, volume 4741 of *Lecture Notes in Computer Science*, pages 256–270. Springer, 2007.

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

[22] Andrew Drucker. New limits to classical and quantum instance compression. *SIAM J. Comput.*, 44(5):1443–1479, 2015.

[23] Paul Erdős and Richard Rado. Intersection theorems for systems of sets. *Journal of the London Mathematical Society*, s1-35(1):85–90, 1960.

[24] Guy Even, Joseph Naor, Satish Rao, and Baruch Schieber. Divide-and-conquer approximation algorithms via spreading metrics. *J. ACM*, 47(4):585–616, 2000.

[25] Stefan Fafianie, Stefan Kratsch, and Vuong Anh Quyen. Preprocessing under uncertainty. In Ollinger and Vollmer [63], pages 33:1–33:13.

[26] Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. Bipartite perfect matching is in quasi-NC. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 754–763. ACM, 2016.

[27] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series)*. Springer, March 2006.

[28] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *J. ACM*, 63(4):29:1–29:60, 2016.

[29] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Bidimensionality and kernels. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 503–510. SIAM, 2010.

[30] Fedor V. Fomin, Saket Saurabh, and Yngve Villanger. A polynomial kernel for proper interval vertex deletion. *SIAM J. Discrete Math.*, 27(4):1964–1976, 2013.

[31] Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct PCPs for NP. *J. Comput. Syst. Sci.*, 77(1):91–106, 2011.

[32] Shivam Garg and Geevarghese Philip. Raising the bar for vertex cover: Fixed-parameter tractability above A higher guarantee. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1152–1166. SIAM, 2016.

[33] Georg Gottlob and Stefan Szeider. Fixed-parameter algorithms for artificial intelligence, constraint satisfaction and database problems. *Comput. J.*, 51(3):303–325, 2008.

[34] Sylvain Guillemot. FPT algorithms for path-transversal and cycle-transversal problems. *Discrete Optimization*, 8(1):61–71, 2011.

[35] Jiong Guo, Hannes Moser, and Rolf Niedermeier. Iterative compression for exactly solving NP-hard minimization problems. In Jürgen Lerner, Dorothea Wagner, and Katharina Anna Zweig, editors, *Algorithmics of Large and Complex Networks*, volume 5515 of *Lecture Notes in Computer Science*, pages 65–80. Springer, 2009.

[36] Rohit Gurjar and Thomas Thierauf. Linear matroid intersection is in quasi-NC. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 821–830. ACM, 2017.

[37] Gregory Z. Gutin, M. S. Ramanujan, Felix Reidl, and Magnus Wahlström. Path-contractions, edge deletions and connectivity preservation. *J. Comput. Syst. Sci.*, 101:1–20, 2019.

[38] Danny Harnik and Moni Naor. On the compressibility of $\mathcal{NP}$ instances and cryptographic applications. *SIAM J. Comput.*, 39(5):1667–1713, 2010.

[39] Danny Hermelin and Xi Wu. Weak compositions and their applications to polynomial lower bounds for kernelization. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 104–113. SIAM, 2012.

[40] Eva-Maria C. Hols and Stefan Kratsch. A randomized polynomial kernel for subset feedback vertex set. In Ollinger and Vollmer [63], pages 43:1–43:14.

[41] Yoichi Iwata, Magnus Wahlström, and Yuichi Yoshida. Half-integrality, LP-branching, and FPT algorithms. *SIAM J. Comput.*, 45(4):1377–1411, 2016.

[42] 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.

[43] Subhash Khot. On the power of unique 2-prover 1-round games. In *IEEE Conference on Computational Complexity*, page 25, 2002.

[44] Stephan Kottler, Michael Kaufmann, and Carsten Sinz. Computation of renameable Horn backdoors. In Hans Kleine Büning and Xishun Zhao, editors, *SAT*, volume 4996 of *Lecture Notes in Computer Science*, pages 154–160. Springer, 2008.

[45] Stefan Kratsch. A randomized polynomial kernelization for vertex cover with a smaller parameter. *SIAM J. Discrete Math.*, 32(3):1806–1839, 2018.

[46] Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In *FOCS*, pages 450–459. IEEE Computer Society, 2012.

[47] Stefan Kratsch and Magnus Wahlström. Compression via matroids: A randomized polynomial kernel for odd cycle transversal. *ACM Transactions on Algorithms*, 10(4):20, 2014.

[48] Frank Thomson Leighton and Ankur Moitra. Extensions and limits to vertex sparsification. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 47–56. ACM, 2010.

[49] Daniel Lokshtanov, Pranabendu Misra, Fahad Panolan, and Saket Saurabh. Deterministic truncation of linear matroids. *ACM Trans. Algorithms*, 14(2):14:1–14:20, 2018.

[50] Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Faster parameterized algorithms using linear programming. *ACM Trans. Algorithms*, 11(2):15:1–15:31, 2014.

[51] László Lovász. Flats in matroids and geometric graphs. In *Proc. Sixth British Combinatorial Conf.*, Combinatorial Surveys, pages 45–86, 1977.

[52] Dániel Marx. Parameterized graph separation problems. *Theor. Comput. Sci.*, 351(3):394–406, 2006.

[53] Dániel Marx. A parameterized view on matroid optimization problems. *Theor. Comput. Sci.*, 410(44):4471–4479, 2009.

[54] Dániel Marx and Igor Razgon. Fixed-parameter tractability of multicut parameterized by the size of the cutset. *SIAM J. Comput.*, 43(2):355–388, 2014.

[55] Dániel Marx and Ildikó Schlotter. Stable assignment with couples: Parameterized complexity and local search. *Discrete Optimization*, 8(1):25–40, 2011.

[56] Syed Mohammad Meesum, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Rank vertex cover as a natural problem for algebraic compression. *SIAM J. Discrete Math.*, 33(3):1277–1296, 2019.

[57] Karl Menger. Zur allgemeinen Kurventheorie. *Fundamenta Mathematicae*, 10(1):96–115, 1927.

[58] Sounaka Mishra, Venkatesh Raman, Saket Saurabh, Somnath Sikdar, and C. Subramanian. The complexity of König subgraph problems and above-guarantee vertex cover. *Algorithmica*, 61:857–881, 2011. 10.1007/s00453-010-9412-2.

[59] Pranabendu Misra, Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. Linear representation of transversal matroids and gammoids parameterized by rank. In Yixin Cao and Jianer Chen, editors, *Computing and Combinatorics - 23rd International Conference, COCOON 2017, Hong Kong, China, August 3-5, 2017, Proceedings*, volume 10392 of *Lecture Notes in Computer Science*, pages 420–432. Springer, 2017.

[60] Ankur Moitra. Approximation algorithms for multicommodity-type problems with guarantees independent of the graph size. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 3–12. IEEE Computer Society, 2009.

[61] N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. LP can be a cure for parameterized problems. In Christoph Dürr and Thomas Wilke, editors, *STACS*, volume 14 of *LIPIcs*, pages 338–349. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.

[62] George L. Nemhauser and Leslie E. Trotter Jr. Vertex packings: structural properties and algorithms. *Math. Program.*, 8:232–248, 1975.

[63] Nicolas Ollinger and Heribert Vollmer, editors. *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, volume 47 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[64] James Oxley. *Matroid Theory*. Oxford University Press, 2011.

[65] Hazel Perfect. Applications of Menger's graph theorem. *J. Math. Anal. Appl.*, 22:96–111, 1968.

[66] Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Paths, flowers and vertex cover. In Camil Demetrescu and Magnús M. Halldórsson, editors, *ESA*, volume 6942 of *Lecture Notes in Computer Science*, pages 382–393. Springer, 2011.

[67] Igor Razgon and Barry O'Sullivan. Almost 2-SAT is fixed-parameter tractable. *J. Comput. Syst. Sci.*, 75(8):435–450, 2009.

[68] Bruce A. Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Oper. Res. Lett.*, 32(4):299–301, 2004.

[69] Neil Robertson and Paul D. Seymour. Graph minors. XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1):65 – 110, 1995.

[70] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003.

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

[72] Magnus Wahlström. Abusing the Tutte matrix: An algebraic instance compression for the K-set-cycle problem. In Natacha Portier and Thomas Wilke, editors, *30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013, February 27 - March 2, 2013, Kiel, Germany*, volume 20 of *LIPIcs*, pages 341–352. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

[73] Sergei Winitzki. *Linear Algebra via Exterior Products*. 2010.

A Problem definitions

This section contains the definitions of all problems encountered in this work.

ALMOST 2-SAT(k)	**Parameter:** k.

Input: A formula $\mathcal{F}$ in 2-CNF and an integer $k \in \mathbb{N}$.
Question: Is there a deletion set X for $\mathcal{F}$ of size at most k?

ALMOST 2-SAT COMPRESSION($|X|$) — **Parameter:** $|X|$.
Input: A 2-CNF formula $\mathcal{F}$, an integer k, and a deletion set X for $\mathcal{F}$.
Question: Is there a deletion set X' for $\mathcal{F}$ of size at most k?

DIGRAPH PAIR CUT(k) — DPC(k) — **Parameter:** k.
Input: A directed graph $D = (V, A)$ with source vertex $s \in V$, a set of pairs $\mathcal{P} \subseteq \binom{V}{2}$, and an integer $k \in \mathbb{N}$.
Question: Is there a set $X \subseteq V \setminus \{s\}$ with $|X| \leq k$ such that no pair in $\mathcal{P}$ is reachable in $D - X$, i.e., such that for each $\{u, v\} \in \mathcal{P}$ at least one of u and v is not reachable from s in $D - X$?

Γ-FEEDBACK VERTEX SET(k) — Γ-FVS(k) — **Parameter:** k.
Input: A directed graph $D = (V, A)$, an edge labeling $\phi : A \to \Gamma$, and an integer k.
Question: Is there a set X of at most k vertices such that $D - X$ has a consistent labeling, i.e., a function $\pi : V \setminus X \to \Gamma$ such that $\pi(u) \otimes \phi((u, v)) = \pi(v)$ for all arcs (u, v) of $D - X$.

KŐNIG VERTEX DELETION ON GRAPHS WITH PERFECT MATCHING(k) — **Parameter:** k.
Input: An undirected graph $G = (V, E)$ that has a perfect matching and an integer k.
Question: Is there a set $X \subseteq V$ of size at most k such that $G - X$ is a Kőnig graph?

s-MULTICUT(k) — **Parameter:** k.
Input: An undirected graph $G = (V, E)$, a family $\mathcal{T} \subseteq \binom{V}{2}$ with $|\mathcal{T}| \leq s$, and $k \in \mathbb{N}$.
Question: Is there a set $X \subseteq V \setminus V(\mathcal{T})$ such that in $G - X$ no component contains both vertices t_1, t_2 of any pair $\{t_1, t_2\} \in \mathcal{T}$?

s-MULTIWAY CUT(k) — **Parameter:** k.
Input: An undirected graph $G = (V, E)$, a set $T \subseteq V$ of at most s terminals, and an integer $k \in \mathbb{N}$.
Question: Is there a set $X \subseteq V \setminus T$ such that each connected component of $G - X$ contains at most one terminal?

RHORN-BACKDOOR DELETION SET(k) — **Parameter:** k.
Input: A CNF-formula $\mathcal{F}$ and an integer $k \in \mathbb{N}$.
Question: Is there a set X of at most k variables such that after deleting from $\mathcal{F}$ all clauses containing a variable of X, the remaining formula $\mathcal{F}'$ is *renamable Horn*, i.e., such that the variables of $\mathcal{F}'$ can be negated so that every clause contains at most one positive literal?

VERTEX COVER ABOVE LP — **Parameter:** $k - |x|$.
Input: An undirected graph $G = (V, E)$, an integer k, and an optimal fractional vertex cover $x \colon V \to [0, 1]$ of total cost denoted by $|x|$.
Question: Is there a set $S \subseteq V$ of size at most k that intersects all edge of G?

VERTEX COVER ABOVE MATCHING — **Parameter:** $k - |M|$.
Input: An undirected graph $G = (V, E)$, an integer k, and a maximum matching M of G.
Question: Is there a set $S \subseteq V$ of size at most k that intersects all edge of G?

VERTEX COVER PARAMETERIZED BY KŐNIG DELETION SET **Parameter:** $|X|$.
Input: An undirected graph $G = (V, E)$, an integer k, and a set $X \subseteq V$ such that $G - X$ is a Kőnig graph.
Question: Is there a set $S \subseteq V$ of size at most k that intersects all edge of G?