Truly Supercritical Trade-offs for Resolution, Cutting Planes, Monotone Circuits, and Weisfeiler–Leman

Susanna F. de Rezende
Lund University

Noah Fleming
Memorial University

Duri Andrea Janett
University of Copenhagen
and Lund University

Jakob Nordström
University of Copenhagen
and Lund University

Shuo Pang
University of Copenhagen

November 21, 2024

Abstract

We exhibit supercritical trade-off for monotone circuits, showing that there are functions computable by small circuits for which any circuit must have depth super-linear or even super-polynomial in the number of variables, far exceeding the linear worst-case upper bound. We obtain similar trade-offs in proof complexity, where we establish the first size-depth trade-offs for cutting planes and resolution that are truly supercritical, i.e., in terms of formula size rather than number of variables, and we also show supercritical trade-offs between width and size for treelike resolution.

Our results build on a new supercritical width-depth trade-off for resolution, obtained by refining and strengthening the compression scheme for the Cop-Robber game in [Grohe, Lichter, Neuen & Schweitzer 2023]. This yields robust supercritical trade-offs for dimension versus iteration number in the Weisfeiler–Leman algorithm, which also translate into trade-offs between number of variables and quantifier depth in first-order logic. Our other results follow from improved lifting theorems that might be of independent interest.

Contents

1   Introduction

Computational complexity aims to understand the amount of resources—such as running time or memory—required in order to solve computational problems. An important regime is to understand how sets of resources interact: can they be optimized simultaneously or are there problems where there is necessarily a trade-off, when optimizing one resource leads to a substantial increase in the others? Traditionally, the strongest trade-offs between two complexity measures, say μ and ν, have been of the following form: it is possible to solve the problem with a small value for μ and with a small value for ν, but optimizing μ causes ν to increase to nearly the value obtained from the brute-force worst-case algorithm (see Figure 1a). In this setting, *robust* trade-offs have been established, where we cannot even approximately optimize μ without a blow-up for ν (corresponding to a tall infeasible region in Figure 1).

Razborov [Raz16] and the earlier works [BBI16, BNT13] show that trade-offs exist which go far beyond this regime, where optimizing one measure causes the other to increase *beyond its worst-case value* (see Figure 1b). These *supercritical* trade-offs have mostly appeared in proof complexity [BBI16, Ber12, BNT13, Raz16, Raz17, Raz18, BN20, FPR22, BT24, CD24] and finite model theory [BN23, GLN23]. Recent papers [GGKS20, FGI+21, FPR22] have raised the question of whether there are supercritical trade-offs in circuit complexity, which brings us to the main contribution of this paper.
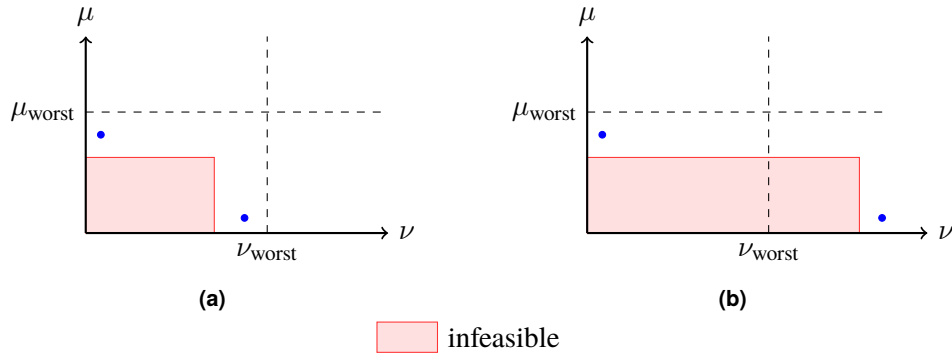

Figure 1: An illustration of trade-offs. Blue dots represent provable upper bounds on measures μ and ν. Proofs with measures in the shaded region are ruled out by the trade-off, where μ_{worst} and ν_{worst} are the worst-case upper bound on μ and ν, respectively. Figure 1a illustrates a non-supercritical trade-off and Figure 1b illustrates a supercritical one.

1.1   Supercritical Trade-offs in Circuit Complexity

Perfect matching is one of the classical problem in complexity theory. Although it has been known to be solvable in polynomial time for nearly 70 years [Edm65], many questions about its computational complexity remain unresolved, in particular regarding its monotone complexity. In a breakthrough result in 1985, Razborov [Raz85] proved the first super-polynomial size lower bound for monotone circuits—Boolean circuits with only AND and OR gates—for two functions: k-clique and bipartite perfect matching. (Independently, Andreev [And85] showed an exponential size lower bound for a different function.) A few years later, Alon and Boppana [AB87] improved the lower bound for k-clique to exponential for large k.

For bipartite perfect matching, however, we still have no better size lower bounds. Raz and Wigderson [RW92] proved a depth-$\Omega(n)$ lower bound, where n is the number of vertices of the graph and the function has $\Theta(n^2)$ inputs. This lower bound is tight, as there are monotone circuits that compute bipartite perfect matching in depth $\mathrm{O}(n)$ (and size $2^{\mathrm{O}(n)}$). In fact, this rather straightforward upper bound remains the best known to this day. Are there monotone circuits computing bipartite perfect matching in size $n^{\mathrm{O}(\log n)}$? If so, why have we not yet been able to find them? And if not, why have we not been able to prove a stronger lower bound?

One possible answer to these questions could be that we have not been able to prove exponential lower bounds because they are simply not true, and that we have not been able to find smaller monotone circuits

computing perfect matching because *they look different*. We know already that if there is a monotone circuit of size $n^{O(\log n)}$ that computes bipartite perfect matching then it must have depth at least $\Omega(n)$. But what if any monotone circuit of size $n^{O(\log n)}$ requires even larger depth, say depth $n^{\Omega(\log n)}$? This could sound like an absurd hypothesis—how can a small circuit require super-linear depth? It is natural to ask, as was done in [GGKS20, FGI$^+$21, FPR22], if there are any monotone functions that exhibit this kind of supercritical trade-off behavior, where small circuits exist but any small circuit requires super-linear depth. We prove this is the case, even for the stronger model of monotone real circuits, where gates can compute any monotone function from two real numbers to a real number.

Theorem 1.1 (Monotone circuit trade-offs). *There are N-variate Boolean functions f_N with either of the following properties:*

(1) f_N is computable by monotone circuits of size $s = \text{poly}(N)$, but any monotone (real) circuit computing f_N of size at most $s^{1.4}$ must have depth at least $N^{2.4}$.

(2) f_N is computable by monotone circuits of size $s = \text{quasipoly}(N)$, but any monotone (real) circuit computing f_N of size at most $s \cdot \exp\big((\log N)^{1.9}\big)$ must have depth super-polynomial in N.

The functions we present that exhibit this behavior actually come from new supercritical trade-offs in the neighboring field of proof complexity.

1.2 Supercritical Trade-offs in Proof Complexity

The *Tseitin formulas*—unsatisfiable systems of $\text{mod } 2$ linear equations—were used to prove the first proof complexity lower bounds in [Tse68]. Since then, these formulas have played a central role in establishing lower bounds for, and understanding the reasoning power of proof systems; see [FP22] for a survey. A notable exception is the *cutting planes* proof system, which captures the types of reasoning achievable when memory is limited to halfspaces. The original paper on cutting planes [Chv73] conjectured that the Tseitin formulas were hard to prove in cutting planes, and this was reiterated in [BP98, Juk12]. While lower bounds on the size of cutting planes proofs have appeared for a variety of formulas [Pud97, BPR95, FPPR22, HP17, Sok24, GGKS20], this conjecture remained open.

In a surprising turn of events, Dadush and Tiwari [DT20] exhibited short (quasi-polynomial size) cutting planes proofs of the Tseitin formulas. Notably, these proofs also have quasi-polynomial depth, far exceeding the linear worst-case upper bound. This raised the question of whether the depth of *any* small cutting planes proof of the Tseitin formulas must be supercritical [BFI$^+$18, FGI$^+$21, FPR22], which would give a partial explanation as to why these proofs took so long to find.

Progress on this question was made in [BGH$^+$06, FGI$^+$21], by showing that any cutting planes proof of the Tseitin formulas on n variables requires depth $\Omega(n)$; and in [FPR22, BT24], by constructing families of CNF formulas which exhibit supercritical size-depth trade-offs for cutting planes. The latter result is somewhat unsatisfactory, however, as the trade-off is supercritical only in the number of variables and not in the size of the formula. This differs from the upper bound in [DT20], which is supercritical in terms of the formula size as well. We refer to trade-offs that are supercritical in the input size—rather than in the number of variables—as *truly* supercritical. In this work, we give the first size-depth trade-offs for cutting planes that are truly supercritical.

Theorem 1.2 (Cutting planes trade-offs). *There are 3-CNF formulas F_N of size $S(F_N)$ over N variables with either of the following properties:*

(1) Resolution refutes F_N in size $S(F_N \vdash \perp) = \text{poly}(N)$, but any cutting planes refutation of size at most $S(F_N \vdash \perp)^{1.4}$ has depth at least $S(F_N)^{2.4}$.

(2) Resolution refutes F_N in size $S(F_N \vdash \perp) = \text{quasi-poly}(N)$, but any cutting planes refutation of size at most $S(F_N \vdash \perp) \cdot \exp\big((\log N)^{1.9}\big)$ has depth at least super-polynomial in N.

As cutting planes simulates resolution, this also implies the first truly supercritical size-depth trade-offs for resolution. In fact, most trade-offs in proof complexity so far are not truly supercritical (with the

exception of [BBI16, Ber12, BNT13]). In this work, we also obtain truly supercritical trade-offs for other combinations of complexity measures, which we state next.

The seminal work [Raz16] provides formulas for which any low-width treelike resolution proof must have size that is *doubly-exponential* in the number of variables. Again, the lower bound is not supercritical in terms of the formula size. We establish a truly supercritical width-size trade-off for treelike resolution.

Theorem 1.3 (Width-size trade-offs). *There are CNF formulas F_N of size $S(F_N) = \text{poly}(N)$ over N variables with either of the following properties:*

(1) Resolution refutes F_N in width $W(F_N \vdash \perp) = o\left(\log N\right)$, but any treelike refutation of width at most $1.4W(F_N \vdash \perp)$ has size at least $\exp\left(S(F_N)^{2.4}\right)$.

(2) Resolution refutes F_N in width $W(F_N \vdash \perp) = o\left((\log N)^{3/2}\right)$, but any treelike resolution refutation of width at most $W(F_n \vdash \perp) + 40\frac{\log N}{\log \log N}$ has size at least $\exp\left(S(F_N)^{\omega(1)}\right)$.

Underlying each of these results is the first truly supercritical width-depth trade-off that is non-trivially robust; the aforementioned results then follow by applying several (new or improved) lifting theorems. Prior to our work, the only truly supercritical trade-off for width versus depth was due to Berkholz [Ber12]; however, this trade-off has no robustness—it holds only for the minimum width—and it therefore cannot be used to obtain other supercritical trade-offs.

Theorem 1.4 (Width-depth trade-offs). *For any constants C and $\delta \in (0, 1)$, there are 4-CNF formulas F_N of size $S(F_N) = \Theta(N)$ over N variables which have resolution refutations of width $w = \lfloor \frac{n}{2 \ln n} \rfloor + 3$, with either of the following properties:*

(1) $N = \text{poly}(n)$ and any refutation of width at most $w + C$ has depth exponential in $\text{poly}(S(F_N))$.

(2) $N = o(2^{n/2})$ and any refutation of width at most $(1 + \delta)w$ has depth super-linear in $S(F_N)$.

1.3 Trade-offs for the Weisfeiler–Leman Algorithm

Surprisingly, all of the results above are obtained by studying the well-known *Weisfeiler–Leman algorithm* for classifying graphs and, more generally, relational structures. This algorithm appears as a subroutine in Babai's celebrated graph isomorphism result [Bab16], and has also been connected to machine learning [MRF+19, Gro21, MLM+23] and many other areas [Kie20, GLNS23]. The 1-dimensional version of the algorithm applied to graphs, known as *color refinement*, starts by coloring all vertices according to their degree. This coloring is then iteratively refined by distinguishing vertices if their multisets of neighborhood colors differ. The process stops when a *stable* coloring is reached, i.e., no further pair of vertices of the same color gets different colors. The *k-dimensional* version of the algorithm (k-WL) instead performs colorings of k-tuples of vertices, or of elements in more general relational structures. Another parameter of interest is the *iteration number*, which is the number of refinement steps until the coloring stabilizes.

It is easy to see that the iteration number of k-WL is at most $n^k - 1$, and this can be slightly improved [KS19, LPS19, GLN23]. For a long time, the best lower bound was linear [Für01] until the sequence of works [BN23, GLN23] showed that $n^{\Omega(k)}$ iterations can be necessary. These results are actually slightly stronger in that they provide robust trade-offs between dimension and iteration number, but they only hold for relational structures of much higher arity than graphs. A stronger $\Omega(n^{k/2})$ lower bound was finally proven in [GLNS23] for pairs of graphs distinguishable in dimension k, but the authors left it as an open problem to turn this into a robust trade-off. Such a result is the starting point of our work.

Theorem 1.5 (Weisfeiler–Leman trade-offs). *For all c, k with $1 \le c \le k - 1$, and for n large enough, there are pairs of graphs of size n that can be distinguished by k-dimensional Weisfeiler–Leman, but for which even $(k + c - 1)$-dimensional Weisfeiler–Leman requires $\Omega\left(n^{k/(c+1)}\right)$ iterations.*

By the well-known equivalence between Weisfeiler–Leman and fragments of first order logic with counting [CFI92], our result also implies trade-offs between variable number and quantifier depth there.
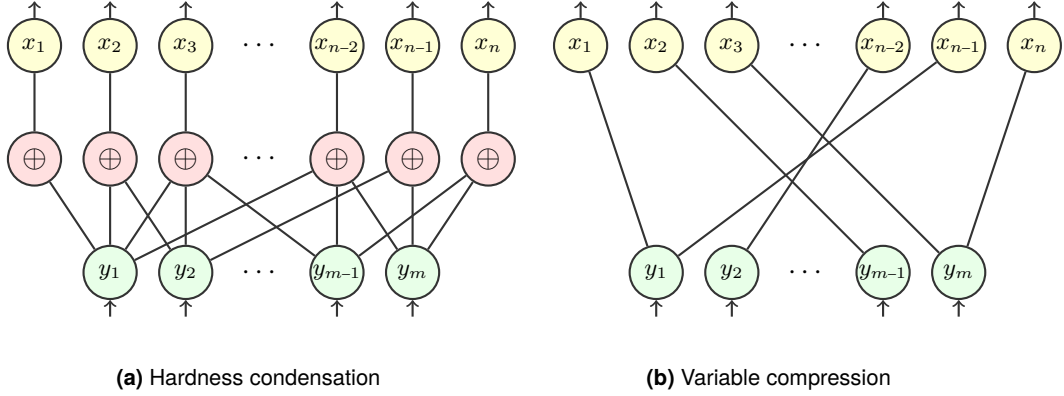
(a) Hardness condensation   **(b)** Variable compression

Figure 2: Hardness condensation in Figure 2a substitutes the x-variables with an XOR over some y-variables, while variable compression in Figure 2b substitutes with y-variables directly. Note that $m \ll n$.

1.4 Techniques

Most of the previously known supercritical trade-offs are based on *hardness condensation* [Raz16], which works by substituting the variables of a problem instance with XOR gadgets over a much smaller set of variables (cf. Figure 2a), and then showing that the substituted instance remains essentially as hard, although the number of variables has decreased substantially. This technique transferred from proof complexity to finite model theory in [BN23] to prove the Weisfeiler–Leman trade-offs discussed above.

The recent result [GLNS23] instead relies on a new technique of *graph compression*, where vertices are identified via an equivalence relation, together with the standard approach of analyzing Weisfeiler–Leman via the *Cop-Robber game* [ST93]. Here, dimension corresponds to number of Cops in play, and iteration number to (game-)rounds. Lower bounds for Weisfeiler–Leman follow from strong Robber strategies, and the bounds become supercritical when these strategies continue to work even when the game is played on the compressed graph. In proof complexity, the number of Cops and rounds approximately correspond to resolution width and depth for Tseitin formulas [GTT20]. Because the correspondence is not exact, [GLNS23] does not give proof complexity results.

Using a refined graph compression and analysis, we obtain (Theorem 1.5) Weisfeiler–Leman trade-offs which are robust. Thanks to this robustness, we are able to translate these results into truly supercritical width-depth trade-offs for resolution, exporting the technique of [GLNS23] to proof complexity, as advocated in that paper. In contrast to hardness condensation, the resulting *compressed* Tseitin formula is obtained by substituting each variable with one of the new variables in a structured way (see Figure 2b). We believe that the tool of *variable compression*, interesting in its own right, may find more applications in proof complexity.

The remaining trade-offs in this paper are obtained by proving and applying new *lifting theorems* to our width-depth trade-off. Lifting is a framework for deriving lower bounds for stronger computation models from those for weak ones. In particular, the lifting theorems of [GGKS20, LMM$^+$22] convert lower bounds on resolution width to lower bounds on the size of monotone (real) circuits, which in turn imply lower bounds for cutting planes. However, the parameters of these theorems are insufficient to obtain supercritical trade-offs from Theorem 1.4. We therefore establish an improved, *tight* lifting theorem for both monotone circuits and cutting planes. The key to the proof is a new way of approximating a combinatorial triangle by structured rectangles, from which we can extract clauses. We also provide an even tighter lifting for resolution size, which has a simple proof based on random restriction. Lastly, we prove a lifting theorem for treelike resolution that turns a depth lower bound into a size lower bound and simultaneously increases the width. We believe that these lifting results should be of independent interest.

4

1.5   Related Work

In concurrent work, Göös, Maystre, Risse and Sokolov [GMRS24] report supercritical size-depth trade-offs for monotone circuits, resolution and cutting planes. Their approach is similar in that they also start with a truly supercritical width-depth trade-off and apply lifting to obtain size-depth trade-offs, but their width-depth trade-off is very different from ours, and relies on a novel, interesting formula construction.

In terms of parameters, their formulas have resolution proofs in width $O(\log n)$, but any proof in width up to n^ϵ has supercritical depth, making their width-depth trade-off extremely robust. This robustness allows them to apply existing lifting theorems as a black box to obtain functions that are computable by monotone circuits of size $n^{O(\log n)}$ but where any monotone circuit of polynomial depth has exponential size. While our results are not nearly as robust, we obtain a blow-up in size even for circuits with depth polynomial in the size upper bound, and our proof complexity trade-offs apply for constant-width proofs. In this sense, the two are incomparable. In addition, we give results for the Weisfeiler–Leman algorithm and prove tight lifting theorems.

Our Weisfeiler–Leman and resolution width-depth trade-offs were announced at the Oberwolfach workshop *Proof Complexity and Beyond* in March 2024. Building on that work, Berkholz, Lichter and Vinall-Smeeth [BLVS24] also obtained a truly supercritical width-size trade-off for treelike resolution. Our treelike resolution trade-off came afterward, with a different technique and improved parameters.

1.6   Organisation of This Paper

This rest of this paper is structured as follows. The preliminaries and a proof overview are presented in Section 2. In Section 3, we define the compressed Cop-Robber game and show how it relates to resolution. In Section 4, we prove the round lower bound on the game. We then prove our lifting theorem for treelike resolution in Section 5 and for resolution in Section 6. In Section 7 we prove our tight general lifting theorem and finally, in Section 8, we discuss some open problems.

2   Preliminaries and Proof Overview

In this section, we present an overview of the components needed to obtain our trade-off results stated in Section 1 and explain how they fit together. We begin with some notation and general definitions.

For $a, b \in \mathbb{N}^+$ with $a \le b$, we use the notation $[a, b] \coloneqq \{a, a+1, \ldots, b\}$, and $[a] = \{1, \ldots, a\}$. Given $k \in \mathbb{N}^+$ and $a, b \in [k]$ with $a > b$, we write $[a, b] \coloneqq \{a, a + 1, \ldots, k\} \cup \{1, 2, \ldots, b\}$. We call sets $[a, b]$, where $a, b \in [k]$, *cyclic intervals modulo* k. In this paper, $\log(\cdot)$ are base 2 and $\ln(\cdot)$ are base e.

All graphs in this paper are simple. For a graph $G = (V, E)$ and a vertex subset $W \subseteq V$, $G|_W$ denotes the induced subgraph on W. Given $F \subseteq E$, we write $V(F)$ for the set of all vertices incident to an edge in F. By a path in a graph, we always mean *simple* paths, i.e., a sequence of distinct vertices where consecutive ones are connected by an edge.

We begin by introducing the Cop-Robber game [ST93], which underpins the proof of our width-depth and Weisfeiler–Leman trade-offs.

2.1   The Cop-Robber Game

We describe the rect$k$-*Cop-Robber game*. The Cops and the Robber stay on vertices of a graph G and can see each other. Initially, the Robber is at a vertex, and all k Cops are lifted from the graph (in a helicopter). A game round unfolds as follows:

(1) If there is no lifted Cop, the Cops choose and lift one. Then they signal a vertex v to the Robber.

(2) The Robber uses a path in G to move from his position w_1 to another vertex w_2 while avoiding the Cops on the graph.

(3) A lifted Cop lands at the signaled vertex v.

The game ends when a Cop lands at the Robber's position.

In Section 3 and Section 4, we analyze a variant called the *compressed Cop-Robber game* [GLNS23]. It is played on a graph added with equivalence relations on vertices and edges, which are obtained as follows. First, a vertex equivalence relation is chosen. For equivalent vertices, the incident edges are then identified one by one, and we take the transitive closure of this identification to get an equivalence relation. In the specific instance we analyze, the graph is a *cylinder* (a grid where columns are cycles) with k rows and roughly n^k columns where vertices on the same row are identified periodically, using a different period in every row. We generalize the construction in [GLNS23] by selecting the row periods based on a parameter $c \leq k - 1$; see Definition 3.4 and Definition 3.9. The original construction corresponds to $c = 2$.

The rules in the compressed game are subtler (see Definition 3.11). In particular, the Robber must avoid *all* vertices equivalent to those occupied by Cops. On the uncompressed cylinder, $k + 1$ Cops have an obvious strategy: block off the middle of the graph—forming a police cordon of sorts—and then march towards the robber in lockstep. With more Cops, and with the compression providing Cop copies on the equivalent vertices, they can potentially do better. Despite that, we prove the following theorem.

Theorem 2.1 (Cop-Robber). *For any parameters $k = k(n)$ and $c = c(n)$ where $1 \leq c \leq k - 1$ and $2 \leq k < n/(2 \ln n)$, there are degree-4 graphs $\{G_n\}$ and a compressed Cop-Robber game on G_n where $k + 1$ Cops can win, but the Robber can survive $\Omega(n^k)$ rounds against $k + c$ Cops.*

The novelty of our analysis, in comparison to [GLNS23], lies in having the Robber play against a virtual stronger opponent, formalized in the notion of *virtual cordons* associated with the Cops (Definition 4.8). The transition of (the set of) virtual cordons over a game round is easier to analyze (see, for example, Lemma 4.11), which allows us to show a strategy for the Robber to survive for $\Omega(n^k)$ rounds.

2.2 The Weisfeiler–Leman Algorithm

We define the Weisfeiler–Leman algorithm on graphs; see the survey [Kie20] for further explanations. A graph $G = (V, E, c)$ where $c \colon V \to \mathbb{N}$ is *vertex colored*. Given $k \geq 2$ and a vertex colored graph $G = (V, E, c)$, the *k-dimensional Weisfeiler–Leman algorithm* [WL68, IL90] iteratively refines a coloring of the k-tuples of vertices. We denote the coloring after the ith round by $\chi^{(i)} \colon V^k \to C$, where C is a finite set. In the initial round, the color $\chi^{(0)}(\vec{u})$ of a tuple $\vec{u} = (u_1, \ldots, u_k)$ is its own isomorphism class, where we say $(u_1, \ldots, u_k)$ is isomorphic to $(v_1, \ldots, v_k)$ if the map $u_i \mapsto v_i$ preserves vertex colors and is an isomorphism between the induced subgraphs of the two tuples. We use $\vec{u}\,[v/u_j]$ to denote the k-tuple obtained by substituting u_j with v in $\vec{u}$, i.e., $(u_1, \ldots, u_{j-1}, v, u_{j+1}, \ldots, u_k)$. In round i, the coloring $\chi^{(i)}(\vec{u})$ of a tuple $\vec{u}$ is obtained by appending a *multiset* of tuples to $\chi^{(i-1)}(\vec{u})$:

$$\chi^{(i)}(\vec{u}) := \left(\chi^{(i-1)}(\vec{u}), \; \left\{\!\!\left\{ \left(\chi^{(i-1)}(\vec{u}\,[v/u_1]), \; \ldots, \; \chi^{(i-1)}(\vec{u}\,[v/u_k]) \right) \;\middle|\; v \in V(G) \right\}\!\!\right\} \right).$$

The algorithm *stabilizes* after round t if any two tuples that have the same color in round t, i.e., $\chi^{(t)}(\vec{u}) = \chi^{(t)}(\vec{v})$, get the same color in round $t + 1$, i.e., $\chi^{(t+1)}(\vec{u}) = \chi^{(t+1)}(\vec{v})$. The minimum such t is called the *iteration number* on G.

The algorithm can be used to distinguish a pair of colored graphs G, H by comparing the colorings $\chi^{(i)}(G)$ and $\chi^{(i)}(H)$. We say that k-dimensional Weisfeiler–Leman *distinguishes G and H in t rounds* if for some color c, the number of tuples that have color c in $\chi^{(t)}(G)$ is different from the number of such tuples in $\chi^{(t)}(H)$.

By applying standard translations (see Appendix A), Theorem 2.1 gives the following trade-off for Weisfeiler–Leman algorithms, which is an explicit version of Theorem 1.5.

Theorem 2.2 (Weisfeiler–Leman trade-offs, explicit). *For all c and k with $1 \leq c \leq k - 1$, if n is large enough, there are n-vertex graph pairs distinguished by k-dimensional Weisfeiler–Leman, but for which $(k + c - 1)$-dimensional Weisfeiler–Leman requires at least $\left(2^{-(c+10)} k^{-3} n \right)^{k/(c+1)}$ iterations.*

Using the equivalence between the k-dimensional Weisfeiler–Leman algorithm and the $(k+1)$-variable fragment of first order logic with counting [CFI92, Theorem 5.2], Theorem 2.2 also implies a trade-off between the number of variables and quantifier depth. Namely, there is a graph pair distinguishable in the $(k+1)$-variable fragment of first order logic with counting, but a lower bound of $\left(2^{-(c+10)}k^{-3}n\right)^{k/(c+1)}$ on the quantifier depth applies up to the $(k+c)$-variable fragment.

2.3 Proof Complexity Basics and Resolution

Let us review some standard definitions from proof complexity. For a more comprehensive presentation of this material, see, e.g., [Kra19, BN21]. A *literal* is a Boolean variable x or its negation $\overline{x}$. It will sometimes be convenient to use the notations $x^1 = x$ and $x^0 = \overline{x}$. A *clause* is a disjunction of literals $D = x_1 \vee \cdots \vee x_k$, which we require to be over pairwise disjoint variables. We call the number of literals appearing in a clause D the *width* $W(D)$ of D. We call a clause of width at most k a k-*clause*. A CNF $F = D_1 \wedge \cdots \wedge D_m$ is a conjunction of clauses, the *formula width* $W(F)$ is the maximal width of clauses in F, the *clause size* $|F|$ is the number of clauses in F (viewed as a set of clauses), and *formula size* $S(F)$ is the sum of width over the clauses in F. We call F a k-*CNF* if all clauses are k-clauses. We denote by $Vars(F)$ the set of variables appearing in a formula F.

A *resolution refutation* $\pi : F \vdash \bot$ of an unsatisfiable CNF formula F is an ordered sequence of clauses $\pi = (D_1, \ldots, D_s)$, where D_s is the empty clause containing no literals denoted by $\bot$, and each D_i is a clause in F, or derived from some specified D_j and D_k, where $j, k < i$, using the *resolution rule*

$$\frac{C \vee x \qquad D \vee \overline{x}}{C \vee D}. \tag{2.1}$$

We associate a DAG G_π with every resolution refutation π as follows. There is a vertex $v_i \in V(G_\pi)$ for every $i \in [s]$, and directed edges $(v_j, v_i), (v_k, v_i) \in E(G_\pi)$ if and only if D_i was derived from D_j and D_k by resolution.

The *size* (or *length*) $S(\pi)$ of a refutation π is the number of clauses s in it. By width $W(\pi)$ of a refutation π, we mean the width of the largest clause in π. Lastly, the *depth* $D(\pi)$ of a refutation π is the number of edges in the longest path in its associated DAG G_π. We also consider the above measures for refuting a CNF formula F, by taking the minimum over all refutations of F. That is, $S(F \vdash \bot) := \min_{\pi:F \vdash \bot}\{S(\pi)\}$, $W(F \vdash \bot) := \min_{\pi:F \vdash \bot}\{W(\pi)\}$, and $D(F \vdash \bot) := \min_{\pi:F \vdash \bot}\{D(\pi)\}$ are the size, width, and depth of refuting F, respectively.

2.4 Supercritical Width-Depth Trade-off for Resolution

Our first technical contribution is the following truly supercritical width-depth trade-off for resolution.

Theorem 2.3 (Width-depth trade-offs, general). *Let $k = k(n)$, $c = c(n)$ be any integer parameters such that $3 \leq c \leq k-1 < \frac{n}{2 \ln n}$. Then for all n, there is a linear-size 4-CNF formula F with between $2k^2n^{c+1}$ and $40k^2(2n)^{c+1}$ variables, which has a resolution refutation of width $k+3$ and size $O(k^2(4n)^k)$, but for which any refutation of width at most $k+c$ has depth at least $\Omega(n^k)$.*

This theorem will be proven through the connection to the compressed Cop-Robber game which we make formal in Section 3. The formula F_N in the theorem is a *Tseitin Formula* [Tse68] after a variable projection operation, also defined in the next section. The Tseitin Formula is defined for any simple graph G where each vertex $v \in V(G)$ is labeled 0 or 1 so that the labels sum to an odd number. A vertex labeled 1 is said to have an *odd charge*. The formula has a variable x_e for every edge $e \in E(G)$ and is defined to be the CNF containing, for all $v \in V(G)$, the clauses expressing that the sum of the edge variables incident to v has parity equal to the label of v.

The two examples in Theorem 1.4 follow from Theorem 2.3 by taking $k(n) := \lfloor n/(2 \log n) \rfloor$ and setting: (1) $c(n)$ to be a large constant, and (2) $c(n)$ to be $\lfloor \frac{1+\delta}{2}k \rfloor$.

2.5 Tight Lifting and Supercritical Trade-offs for Resolution

The framework for obtaining our other proof and circuit complexity trade-offs from the width-depth trade-off is *lifting* which is based on composition with functions, which we refer to as *gadgets*. For CNF formulas, there can be multiple ways of representing its composition with a gadget as a CNF formula. Therefore, for the gadgets g we are interested in, we will denote by $g(F)$ a specific CNF encoding of the composition of the CNF formula F with the gadget g.

In this paper, we consider two gadgets: $\mathrm{XOR}_m : \{0,1\}^m \to \{0,1\}$, defined as $\mathrm{XOR}_m(x_1, \ldots, x_m) = \bigoplus_{i \in [m]} x_i$, and $\mathrm{IND}_m : [m] \times \{0,1\}^m \to \{0,1\}$, defined as $\mathrm{IND}_m(x,y) = y_x$. Given a CNF formula F over variables $x_1, \ldots, x_n$, we denote by $\mathrm{XOR}_m(F)$ the CNF formula obtained by substituting each x_i by $y_{i,1} \oplus \cdots \oplus y_{i,m}$ where $y_{i,j}$ is a new propositional variable, and then expanding it out in CNF. For instance, if $m = 2$ then the clause $x_4 \vee \overline{x}_5$ yields 4 clauses:

$$y_{4,1} \vee y_{4,2} \vee y_{5,1} \vee \overline{y}_{5,2}\,, \qquad y_{4,1} \vee \overline{y}_{4,2} \vee y_{5,1} \vee y_{5,2}\,, \tag{2.2}$$

$$\overline{y}_{4,1} \vee y_{4,2} \vee \overline{y}_{5,1} \vee \overline{y}_{5,2}\,, \qquad \overline{y}_{4,1} \vee y_{4,2} \vee y_{5,1} \vee y_{5,2}\,. \tag{2.3}$$

Note that the width of $\mathrm{XOR}_m(F)$ is $m \cdot W(F)$ and the number of clauses is $|\mathrm{XOR}_m(F)| \le 2^{(m-1) \cdot W(F)}|F|$.

Our lifting theorem for treelike resolution, which we prove in Section 5, uses composition with the XOR_m gadget. Observe that the resolution refutation in its conclusion has small depth and simultaneously smaller width. This decrease in width is essential for obtaining our width-size trade-off.

Theorem 2.4 (Lifting for treelike resolution). *Let F be a CNF formula and let $m \ge 2$. If there is a width-w, size-s treelike resolution refutation for $\mathrm{XOR}_m(F)$, then there is a width-$\left(\frac{w}{m-1}\right)$, depth-$\log s$ resolution refutation of F.*

We can now apply this theorem to our width-depth trade-off to obtain the supercritical trade-offs for treelike resolution.

Theorem 2.5 (Width-size trade-offs, general). *For any $m = m(n) \ge 3$, $k = k(n) \in [4, \frac{n}{2\ln n}]$, and $\varepsilon = \varepsilon(n) \in (\frac{4}{k}, 1 - \frac{1}{k})$, there are $4m$-CNF formulas F_N with N variables and formula size $\mathrm{O}(16^m \cdot N)$, where $2k^2 n^{\lfloor \varepsilon k \rfloor} m \le N \le 40 k^2 (2n)^{\lfloor \varepsilon k \rfloor} m$, which are refutable in width $m(k+3)$ resolution, but for which any treelike refutation of width at most $(m-1)(1+\varepsilon)k$ has size at least $2^{\Omega(n^k)}$.*

Proof. Let F be the 4-CNF formulas from Theorem 2.3 with parameter $c := \lfloor \varepsilon k \rfloor - 1 \in [3, k-1]$, and define $F_N := \mathrm{XOR}_m(F)$. Then $S(F_N) = \mathrm{O}(2^{4(m-1)} \cdot 4m \cdot S(F)) = \mathrm{O}(2^{4m} \cdot m \cdot |\mathit{Vars}(F)|) = \mathrm{O}(16^m \cdot N)$, and since F is refutable in width $k+3$, a line-by-line simulation via $x_i = y_{i,1} \oplus \ldots \oplus y_{i,m}$ gives a refutation of F_N in width $m(k+3)$. Now suppose π is a treelike refutation of F_N in width $(m-1)(1+\varepsilon)k$ and size s, then by Theorem 5.3, there is a refutation of F in width $(1+\varepsilon)k$ and depth $\log s$. The theorem follows since Theorem 2.3 implies that $\log s = \Omega(n^k)$. □

Note that Theorem 1.3 follows immediately from Theorem 2.5 by taking $k(n) := \lfloor n/(2\log n) \rfloor$ and setting: (1) $m := 256$, $\varepsilon := 0.41$, and (2) $m := \lfloor \sqrt{n} \rfloor$, $\varepsilon := \frac{100}{\sqrt{n}}$.

Now, as a warm up for the lifting theorems for monotone circuits and cutting planes in Section 2.6, we prove an even tighter result for resolution. For this theorem, we consider the following composition of a CNF formula with the indexing gadget.[1] Let F be a CNF formula over variables $z_1, \ldots, z_n$. To obtain the CNF formula $\mathrm{IND}_m(F)$, we start with substituting in F every occurrence of z_i by

$$(x_{i,1} \to y_{i,1}) \wedge \ldots \wedge (x_{i,m} \to y_{i,m})\,, \tag{2.4}$$

where $x_{i,j}$ and $y_{i,j}$ are new propositional variables, and we expand it out to CNF. Moreover, we would like to include $x_{i,1} \vee \ldots \vee x_{i,m}$ for each i to ensure that $x_{i,j} = 1$ for at least one $j \in [m]$; but to keep the width of the formula small, we instead use extension variables to encode each of these clauses as a 3-CNF formula with $\le m$ clauses. Note that the width of $\mathrm{IND}_m(F)$ is $2W(F)$ and the number of clauses is $|\mathrm{IND}_m(F)| \le m^{W(F)}|F| + nm$. Using this gadget, we obtain our lifting theorem for resolution.

---

[1]Other standard encodings work as well, but this one ensures the formula width increase by at most a factor 2.

Theorem 2.6 (Lifting for resolution). *For any $m, n \geq 1$ and n-variate CNF formula F, if $\mathrm{IND}_m(F)$ has a resolution refutation of size S and depth d, then F has a resolution refutation of width $\lfloor \log_{(m+1)/2} S \rfloor$ and depth d.*

In this theorem, the size-width relation is nearly tight (see Lemma 2.7 below), and there is no increase in depth. Moreover, the theorem holds for any gadget size, and the proof, which we defer to Section 6, is simple—based on a random restriction argument.

By a standard step-by-step simulation we obtain the following upper bound for refuting $\mathrm{IND}_m(F)$. We include the proof for the sake of completeness.

Lemma 2.7. *For any $m, n \geq 1$ and n-variate CNF formula F, if F has a resolution refutation of width w and size $s \geq n$, then $\mathrm{IND}_m(F)$ has a resolution refutation of size $O(s \cdot m^{w+1})$.*

Proof. The proof is a standard step-by-step simulation. Let F be a CNF formula over variables $z_1, \ldots, z_n$ and let Π be a resolution refutation of F in width w and size s. We start by deriving $\bigvee_{j \in [m]} x_{i,j}$ for all $i \in [n]$ from the axioms in $\mathrm{IND}_m(F)$, which can be done in $O(nm)$ steps. We then simulate Π step by step, keeping the invariant that for every clause $C = \bigvee_{\ell \in [w']} z_{i_\ell}^{\beta_\ell}$ in Π, we derive, for each $J = (j_1, \ldots, j_{w'}) \in [m]^{w'}$, the clause $C_J = \bigvee_{\ell \in [w']} (\overline{x}_{i_\ell, j_\ell} \vee y_{i_\ell, j_\ell}^{\beta_\ell})$. This holds for the axioms by definition of $\mathrm{IND}_m(F)$. Suppose it holds for clause $C \vee z_i$ and $D \vee \overline{z}_i$, and let w' be the width of $D \vee C$. Then for any $J = (j_1, \ldots, j_{w'}) \in [m]^{w'}$ and any $j \in [m]$ we can derive $(D \vee C)_J \vee \overline{x}_{i,j}$ in one step by resolving over variable $y_{i,j}$. Finally, we can derive $(D \vee C)_J$ in m steps by resolving $(D \vee C)_J \vee \overline{x}_{i,j}$ for all $j \in [m]$ with $\bigvee_{j \in [m]} x_{i,j}$. This give a total of $m^{w'+1} + m = O(m^{w+1})$ steps per new clause in Π. $\square$

We can now apply Theorem 2.6 to our width-depth trade-off to obtain supercritical size-depth trade-offs for resolution.

Theorem 2.8 (Resolution size-depth trade-offs, general). *For any $m = m(n)$, $k = k(n)$, and $c = c(n)$ such that $3 \leq c \leq k - 1 < \frac{n}{2 \ln n}$, there are 8-CNF formulas F_n with $O(mk^2(2n)^{c+1})$ variables and formula size $S(F_n) = O(m^4 k^2 (2n)^{c+1})$ which resolution can refute in size $O(m^{k+4} k^2 (4n)^k)$, but for which any refutation of size at most $\left(\frac{m+1}{2}\right)^{k+c}$ has depth at least $\Omega(n^k)$.*

Proof. Let $F_N = \mathrm{IND}_m(F)$, where F is the formula obtained from Theorem 2.3, our supercritical width-depth trade-off, for the parameters c, k and n. Note that F_N is a 8-CNF formula of size $O(m^4 k^2 (2n)^{c+1})$. Since by Theorem 2.3 F has a resolution refutation of width $k + 3$ and size $O(k^2 (4n)^k)$, we have by Lemma 2.7 that $\mathrm{IND}_m(F)$ has a resolution refutation of size $O(m^{k+4} k^2 (4n)^k)$. The lower bounds follows from combining the lifting theorem (Theorem 2.6) and the width-depth trade-off (Theorem 2.3). $\square$

2.6 Tight Lifting and Supercritical Trade-offs for Monotone Circuits and Cutting Planes

A *monotone real circuit* is a Boolean circuit whose gate-set includes all monotone functions of the form $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. It has n input gates $x_1, \ldots, x_n$ and must output a bit in $\{0, 1\}$. Note that monotone real circuits are an extension of traditional monotone circuits.

We define the more general (semantic) version of cutting planes, to which our lower bounds also apply. A *semantic cutting planes refutation* of a system of linear inequalities $Ax \geq b$ is a sequence of inequalities $\{c_i x \geq d_i\}_{i \in [s]}$, with $c_i \in \mathbb{Z}^n$, $d_i \in \mathbb{Z}$, such that the final inequality is the contradiction $0 \geq 1$, and for every $i \in [s]$, $c_i x \geq d_i$ either belongs to $Ax \geq b$ or follows from two previous inequalities by a semantic deduction step, that is, from $ax \geq b$ and $a'x \geq b'$ we can derive any $cx \geq d$ which satisfies $(ax \geq b) \wedge (a'x \geq b') \implies cx \geq d$ for every $x \in \{0, 1\}^n$. The *size* of a semantic cutting planes refutation is s, the number of inequalities in the sequence. One may view a semantic cutting planes proof as a DAG with one vertex per inequality such that the leaves are the inequalities belonging to $Ax \geq b$, the root is $0 \geq 1$, and every non-leaf vertex has two incoming edges the vertices from which it was derived. The *depth* of a semantic cutting planes proof is the longest root-to-leaf path in this DAG.

Like previous DAG lifting theorems, it will be convenient to work with the following top-down definitions of these models—*rectangle-* and *triangle*-DAGs solving *(total) search problems*. A *search*

problem is a relation $\mathcal{S} \subseteq \mathcal{D} \times \mathcal{O}$ where for every input $x \in \mathcal{D}$, there is at least one output $o \in \mathcal{O}$ such that $(x, o) \in \mathcal{S}$. We start by defining *shape-DAGs* [GGKS20], which are a generalisation of rectangle-DAGs introduced in [Raz95] and simplified in [Pud10, Sok17].

Definition 2.9 (Shape-DAG). Let $\mathcal{F} \subseteq \mathcal{D}$ be a family of sets, which we call the "shapes" of the DAG, and $\mathcal{S} \subseteq \mathcal{D} \times \mathcal{O}$ be a search problem. An $\mathcal{F}$-DAG solving $\mathcal{S}$ is a fan-in ≤ 2 rooted directed acyclic graph where each vertex v is labeled with a shape $S_v \in \mathcal{F}$ such that the following hold:

1. *Root.* The distinguished root r is labelled with the "full" shape $S_r = \mathcal{D}$.
2. *Non-Leaves.* If u has children v, w then $S_u \subseteq S_v \cup S_w$.
3. *Leaf.* If ℓ is a leaf of the DAG then there is some $o \in \mathcal{O}$ such that $S_\ell \subseteq \mathcal{S}^{-1}(o)$.

The *size* of an $\mathcal{F}$-DAG is the number of nodes it contains, and the *depth* is the length of the longest root-to-leaf path in the DAG.

For a bipartite input domain $X \times Y$, a *rectangle* $R = R^X \times R^Y$ is a product set, where $R^X \subseteq X$ and $R^Y \subseteq Y$. A *triangle* is a subset $T \subseteq X \times Y$ that can be written as $T = \{(x, y) \mid a_T(x) < b_T(y)\}$ for some labeling of the rows $a_T : X \to \mathbb{R}$ and columns $b_T : Y \to \mathbb{R}$ by real numbers. A *rectangle-DAG* is a shape-DAG where the set of shapes $\mathcal{F}$ is the set of all rectangles over the input domain. Similarly, a *triangle-DAG* is a shape-DAG where $\mathcal{F}$ is the set of all triangles. Note that because any rectangle is also a triangle, a rectangle-DAG is a special case of a triangle-DAG.

We now introduce the two types of search problems that allow us to relate triangle- and rectangle-DAGs to cutting planes and monotone circuits. Let $F = C_1 \wedge \cdots \wedge C_m$ be an unsatisfiable CNF formula on n variables. The *falsified clause search problem for F* is the following total search problem: given $z \in \{0, 1\}^n$, find an $i \in [m]$ such that the clause C_i is falsified by z. Formally, we define the relation $\mathsf{Search}(F) \subseteq \{0, 1\}^n \times [m]$ by

$$(z, i) \in \mathsf{Search}(F) \iff C_i(z) = 0. \tag{2.5}$$

We are sometimes interested in bipartite input domains, so given a partition of the variables of F, where we define the relation $\mathsf{Search}^{X,Y}(F) \subseteq (X \times Y) \times [m]$ by $((x, y), i) \in \mathsf{Search}(F) \iff C_i(z) = 0$. It is not difficult to see that for any CNF formula F and any partition of its variables, a semantic cutting planes refutation of F implies, for any partition of the variables of F, a triangle-DAG for $\mathsf{Search}^{X,Y}(F)$ of the same size and depth; indeed, any halfspace $az \geq b$ defines a triangle $H := \{z \in \{0, 1\}^n \mid az < b\}$. Similarly, a resolution refutation of F implies a rectangle-DAG for $\mathsf{Search}^{X,Y}(F)$ of the same size and depth.

Given a total or partial monotone function $f : \{0, 1\}^n \to \{0, 1\}$, the *monotone Karchmer–Wigderson search problem* [KW90] $\mathsf{mKW}(f) \subseteq (f^{-1}(1) \times f^{-1}(0)) \times [n]$ is defined as

$$((x, y), i) \in \mathsf{mKW}(f) \iff x_i > y_i. \tag{2.6}$$

The DAG-like version of the monotone Karchmer–Wigderson relation [Raz95, Pud10, Sok17] implies that there is a monotone circuit (respectively, monotone real circuit) computing f if and only if there is a rectangle-DAG (respectively, triangle-DAG) solving $\mathsf{mKW}(f)$ of the same size and depth.

For our lifting theorems we need to compose search problems with gadgets. Given a search problem $\mathcal{S} \subseteq \{0, 1\}^n \times \mathcal{O}$ and a gadget $g : \mathcal{D} \to \{0, 1\}$, we can define $\mathcal{S} \circ g^n \subseteq \mathcal{D}^n \times \mathcal{O}$ to be the relation where $(x, o) \in \mathcal{S} \circ g^n$ if and only if $(z, o) \in \mathcal{S}$, where $z_i = g(x_i)$ for $i \in [n]$. We also consider the search problem $\mathsf{Search}^{X,Y}(\mathrm{IND}_m(F))$, where X corresponds to the x-variables, and Y to the y-variables of $\mathrm{IND}_m(F)$. By a standard reduction [Gál01, Raz90], there is a way of translating between the composed search problems; see e.g. [GGKS20] for a proof.

Fact 2.10. *Let F be an unsatisfiable k-CNF on ℓ clauses and n variables, let $m = m(n)$ be a parameter and $N = \ell \cdot (2m)^k$. There is a partial monotone function $f : \{0, 1\}^N \to \{0, 1\}$ such that*

1. $\mathsf{Search}(F) \circ \mathrm{IND}_m^n$ *reduces to* $\mathsf{mKW}(f)$. *In particular, an $\mathcal{F}$-DAG solving* $\mathsf{mKW}(f)$ *implies an $\mathcal{F}$-DAG solving* $\mathsf{Search}(F) \circ \mathrm{IND}_m^n$ *of the same size and depth.*

2. $\mathsf{mKW}(f)$ *reduces to* $\mathsf{Search}^{X,Y}(\mathrm{IND}_m(F))$. *In particular, an $\mathcal{F}$-DAG solving* $\mathsf{Search}^{X,Y}(\mathrm{IND}_m(F))$ *implies an $\mathcal{F}$-DAG solving* $\mathsf{mKW}(f)$ *of the same size and depth.*

We now state our lifting theorem from resolution to triangle-DAGs.

Theorem 2.11 (Lifting for triangle-DAGs). *Let F be an n-variate unsatisfiable CNF formula, and let $m, w \in \mathbb{N}, \delta > 0$ be arbitrary parameters satisfying $w \leq n$, $0 < \delta < 1 - \frac{1}{\log m}$ and $m \geq (\frac{50n}{\delta})^{2/\delta}$. If there is a triangle-DAG of size $\frac{1}{2}m^{(1-\delta)w}$ and depth d solving $\mathsf{Search}(F) \circ \mathrm{IND}_m^n$, then F has a resolution refutation of width w and depth dw.*

We prove this theorem in Section 7. Combining this lifting theorem with our width-depth trade-off for resolution (Theorem 2.3) we obtain the supercritical size-depth trade-offs for monotone (real) circuits.

Theorem 2.12 (Monotone circuit trade-offs, general). *For any integers $c = c(n), k = k(n), m = m(n)$ and real number $\delta = \delta(n) \in (0, 0.9)$ such that $3 \leq c \leq k - 1 < \frac{n}{2\log n}$ and $m \geq (\frac{50n}{\delta})^{2/\delta}$, the following holds for sufficiently large n. There are N-variate functions f_N over $N = \mathrm{O}(m^4 k^2 (2n)^{c+1})$ variables computable by a monotone circuit with size at most $\mathrm{O}(m^{k+4} k^2 (4n)^k)$, but for which any monotone real circuit with size at most $\frac{1}{2}m^{(1-\delta)(k+c)}$ must have depth at least $\Omega(n^k/k)$.*

Proof. Let F be the 4-CNF formula obtained from Theorem 2.3, our supercritical width-depth trade-off, for the parameters c, k and n. Consider the partial monotone function $g_N : \{0,1\}^N \to \{0,1\}$ obtained by applying Fact 2.10 to F. We have that $N = \mathrm{O}(m^4 k^2 (2n)^{c+1})$. Since by Theorem 2.3, F has a resolution refutation of width $k + 3$ and size $O(k^2(4n)^k)$, we have by Lemma 2.7 that $\mathsf{mKW}(g_N)$ can be solved by a rectangle-DAG of size $\mathrm{O}(m^{k+4} k^2 (4n)^k)$, where we use the fact that a resolution refutation of F implies a rectangle-DAG solving $\mathsf{Search}^{X,Y}(\mathrm{IND}_m(F))$ in the same size, and that by Fact 2.10 $\mathsf{mKW}(g_N)$ reduces to $\mathsf{Search}^{X,Y}(\mathrm{IND}_m(F))$. This implies that there is a monotone circuit of the same size computing g_N. Let f_N be the total function, which extends g_N, computed by this circuit.

Now, if there is a monotone real circuit of size s and depth d computing f_N, then there is a triangle-DAG of size s and depth d solving $\mathsf{mKW}(f_N)$, and hence also $\mathsf{mKW}(g_N)$. By Fact 2.10 this implies there is a triangle-DAG solving $\mathsf{Search}(F) \circ \mathrm{IND}_m^n$ in the same size and depth. Finally, combining the triangle-DAG lifting theorem (Theorem 2.11) and the width-depth trade-off (Theorem 2.3) we conclude that if $s \leq \frac{1}{2}m^{(1-\delta)(k+c)}$ then $d = \Omega(n^k/(k+c)) = \Omega(n^k/k)$. $\qquad \square$

We can obtain a similar supercritical trade-off for cutting planes.

Theorem 2.13 (Cutting planes trade-offs, general). *For any integers $c = c(n), k = k(n), m = m(n)$ and real number $\delta = \delta(n) \in (0, 0.9)$ such that $3 \leq c < k < \frac{n}{2\log n}$ and $m \geq (\frac{50n}{\delta})^{2/\delta}$, the following holds for all n. There are unsatisfiable 3-CNF formulas F_N of size $N = \mathrm{O}(m^4 k^2 (2n)^{c+1})$ that can be refuted in resolution in size $\mathrm{O}(m^{k+4} k^2 (4n)^k)$, but for which any semantic cutting planes refutation in size at most $\frac{1}{2}m^{(1-\delta)(k+c)}$ must have depth at least $\Omega(n^k/k)$.*

This theorem can be proven along the same lines as Theorem 2.12, by applying the lifting theorem (Theorem 2.11) to the width-depth trade-off (Theorem 2.3) together with Fact 2.10, and using Lemma 2.7 for the upper bound. The only caveat is that this would give us a 8-CNF formula. In order to obtain a 3-CNF formula, we need to define a 3-CNF version of $\mathrm{IND}_m(F)$, denoted by $\widetilde{\mathrm{IND}}_m(F)$. Let F be a CNF formula over variables $z = z_1, \ldots, z_n$, then the formula $\widetilde{\mathrm{IND}}_m(F)$ is over variables $x_{i,j}$ and $y_{i,j}$ where $i \in [n]$ and $j \in [m]$, the extension variables to write each of the clauses $\bigvee_{j \in [m]} x_{i,j}$, for $i \in [n]$, as a 3-CNF formula, along with variables $x_{C,J}$ and $y_{C,J}$ for every $C \in F$ and every $J \in [m]^{W(C)}$. The clauses in $\widetilde{\mathrm{IND}}_m(F)$ consist of: a 3-CNF encoding of $\bigvee_{j \in [m]} x_{i,j}$ for every $i \in [n]$; for every $C = \bigvee_{\ell \in [w]} z_{i_\ell}^{\beta_\ell}$ in F and every $J = (j_1, \ldots, j_w) \in [m]^w$, a 3-CNF encoding of $(\bigwedge_{\ell \in [w]} x_{i_\ell, j_\ell}) \to x_{C,J}$, a 2-clause $x_{C,J} \to y_{C,J}$, and a 3-CNF encoding of $y_{C,J} \to \bigvee_{\ell \in [w]} y_{i_\ell, j_\ell}^{\beta_\ell}$. Note that if F is a w-CNF formula, then $\widetilde{\mathrm{IND}}_m(F)$ has $\mathrm{O}(w \cdot |F| \cdot m^w + nm)$ variables and clauses.

We observe two basic facts about $\widetilde{\mathrm{IND}}_m(F)$. First, every size-$s$ resolution refutation of $\mathrm{IND}_m(F)$ can be made into a size-$\mathrm{O}(s+|\widetilde{\mathrm{IND}}_m(F)|)$ refutation of $\widetilde{\mathrm{IND}}_m(F)$. This is because $\mathrm{IND}_m(F)$ can be derived from $\widetilde{\mathrm{IND}}_m(F)$ in linear size. Secondly, for both rectangle- and triangle-DAGs (or any shape-DAG that is closed under taking intersection with rectangles), the search problem $\mathsf{Search}^{X,Y}(\mathrm{IND}_m(F))$ reduces to $\mathsf{Search}^{\widetilde{X},\widetilde{Y}}(\widetilde{\mathrm{IND}}_m(F))$, where $\widetilde{X}$ corresponds to the x-variables, and $\widetilde{Y}$ to the y-variables of $\widetilde{\mathrm{IND}}_m(F)$. Indeed, we can fix a pair of injective maps $\phi_X : \{0,1\}^X \to \{0,1\}^{\widetilde{X}}$ and $\phi_Y : \{0,1\}^Y \to \{0,1\}^{\widetilde{Y}}$ which extend every assignment on $X \cup Y$ to one on $\widetilde{X} \cup \widetilde{Y}$ according to the semantic meaning of the new variables. Let $\mathcal{O}$ be the set of possible outputs of $\mathsf{Search}^{X,Y}(\mathrm{IND}_m(F))$, which we view as the set of clauses of $\mathrm{IND}_m(F)$. Similarly, let $\widetilde{\mathcal{O}}$ be the set of clauses of $\widetilde{\mathrm{IND}}_m(F)$. We can define an injective map $\phi_{\widetilde{\mathcal{O}}} : \widetilde{\mathcal{O}} \to \mathcal{O}$ which given a clause in $\widetilde{\mathrm{IND}}_m(F)$ outputs the clause $\mathrm{IND}_m(F)$ it came from. Therefore, given an $\mathcal{F}$-DAG, where $\mathcal{F}$ is a shape-DAG closed under taking intersections with rectangles, $\widetilde{\Gamma}$ solving $\mathsf{Search}^{\widetilde{X},\widetilde{Y}}(\widetilde{\mathrm{IND}}_m(F))$, we can create an $\mathcal{F}$-DAG Γ solving $\mathsf{Search}^{X,Y}(\mathrm{IND}_m(F))$ with the same topology, as follows. For each node in $\widetilde{\Gamma}$—which is a subset of $\{0,1\}^{\widetilde{X}} \times \{0,1\}^{\widetilde{Y}}$—we take its intersection with $\phi_X(\{0,1\}^X) \times \phi_Y(\{0,1\}^Y)$ and view it as a subset of $\{0,1\}^X \times \{0,1\}^Y$ via $\phi_X^{-1} \times \phi_Y^{-1}$, giving the corresponding node of Γ. It is not hard to see that Γ is an $\mathcal{F}$-DAG for $\mathsf{Search}^{X,Y}(\mathrm{IND}_m(F))$.

Proof of Theorem 2.13. Let $F_N = \widetilde{\mathrm{IND}}_m(F)$, where F is the formula obtained from Theorem 2.3, our supercritical width-depth trade-off, for the parameters c, k and n. Note that F_N is a 3-CNF formula that has $\mathrm{O}(m^4 k^2 (2n)^{c+1})$ variables and clauses. Since by Theorem 2.3, F has a resolution refutation of width $k+3$ and size $O(k^2(4n)^k)$, we have by Lemma 2.7 that $\mathrm{IND}_m(F)$, and hence also $\widetilde{\mathrm{IND}}_m(F)$, has a resolution refutation of size $\mathrm{O}(m^{k+4} k^2 (4n)^k)$.

Now, if there is a semantic cutting planes refutation of F_N of size s and depth d, then there is a triangle-DAG solving $\mathsf{Search}^{X,Y}(\mathrm{IND}_m(F))$ of size s and depth d, using the fact above that $\mathsf{Search}^{X,Y}(\mathrm{IND}_m(F))$ reduces to $\mathsf{Search}^{\widetilde{X},\widetilde{Y}}(\widetilde{\mathrm{IND}}_m(F))$. By Fact 2.10 this gives a triangle-DAG solving $\mathsf{Search}(F) \circ \mathrm{IND}_m^n$ of the same size and depth. Finally, combining the triangle-DAG lifting theorem (Theorem 2.11) and the width-depth trade-off (Theorem 2.3), we conclude that if $s \leq \frac{1}{2} m^{(1-\delta)(k+c)}$ then $d = \Omega(n^k/(k+c)) = \Omega(n^k/(k+c))$. $\qquad\square$

Now Theorem 1.1 follows from Theorem 2.12, and Theorem 1.2 from Theorem 2.13, by setting: (1) k to be a sufficiently large constant, $c := \lfloor 0.41k \rfloor$, $w := k + c$, $\delta := \frac{1}{200}$, and $m := n^{500}$; and (2) $k := \lfloor \frac{n}{4 \log n} \rfloor$, $c := \lfloor \sqrt{k} \rfloor$, $w := k + c$, $\delta := \frac{1}{4\sqrt{k}}$, and $m := \lfloor n^{3/\delta} \rfloor = \lfloor n^{12\sqrt{k}} \rfloor$.

3 Compressed Cop-Robber Game and the Formula

In this section, we define the compressed Cop-Robber game introduced by [GLNS23]. In Section 3.1, we define graph compression and the resulting compressed Tseitin formula in general terms, and then construct a concrete instance generalizing the one in [GLNS23]. In Section 3.2, we define the compressed game. In Section 3.3, we show some basic facts about resolution and the compressed game.

3.1 Graph Compression and Compressed Tseitin Formula

In the next definition, we assume that the graph G is given together with its adjacency list. That is, for each vertex there is an ordered list of its neighbors.

Definition 3.1 (Graph compression). We call an equivalence relation $\equiv_V$ on $V(G)$ *compatible* if $u \equiv_V v$ implies that u, v are non-adjacent and have the same degree. A compatible $\equiv_V$ induces an equivalence relation $\equiv_E$ on $E(G)$ as follows. First, we let two edges e_1 and e_2 be equivalent if there are $v_1, v_2 \in V(G)$ such that $e_1 = \{v_1, w_1\}$, $e_2 = \{v_2, w_2\}$, $v_1 \equiv_V v_2$, and the position of w_1 in the neighbor list of v_1 equals the position of w_2 in that of v_2. Then we take the transitive closure of this relation on $E(G)$ to be $\equiv_E$. We call the triple $(G, \equiv_V, \equiv_E)$ a *graph compression*.

Given a graph compression $(G, \equiv_V, \equiv_E)$, let $/_{\equiv_V}$ be the map from the vertices to their equivalence classes, and for $W \subseteq V(G)$, let $W/_{\equiv_V}$ denote the image of W and $W_{\equiv_V} := \{v \mid v \equiv_V w \text{ for some } w \in W\}$. Similarly, let $/_{\equiv_E}$ be the map from the edges to their equivalence classes, and for $F \subseteq E(G)$, let $F/_{\equiv_E}$ denote the image of F and $F_{\equiv_E} := \{e \mid e \equiv_E f \text{ for some } f \in F\}$. We write $\{e\}_{\equiv_E}$ as $e_{\equiv_E}$ and $\{v\}_{\equiv_V}$ as $v_{\equiv_V}$.

Definition 3.2 (Compressed Tseitin). Given a Tseitin formula on G and a graph compression $(G, \equiv_V, \equiv_E)$, the edge equivalence $\equiv_E$ induces a variable substitution $x_e \mapsto x_{e/\equiv_E}$ as follows: for each equivalence class, introduce a single, new variable $x_{e/\equiv_E}$ and replace the variable of every edge in this equivalence class by $x_{e/\equiv_E}$. We call the resulting CNF formula the *compressed Tseitin formula*.

Remark 3.3. Observe that the compressed Tseitin formula has width at most $\deg(G)$, the maximal vertex degree, and $|E(G)/\equiv_E|$-many variables. Also, the parity constraints at equivalent vertices become the same after the substitution, and so the compressed formula has at most $2^{\deg(G)-1}|V(G)/\equiv_V|$-many clauses.

The formula remains unsatisfiable since it is obtained from an unsatisfiable formula via a variable substitution. Moreover, since this substitution is a projection (i.e., each x_e is substituted with one variable), any resolution refutation of the original formula gives rise to one of the compressed formula with no greater proof width, depth, or size.

We will focus on an explicit graph and graph compression. For a fixed $k \geq 2$, we define G_{cyl} to be the *cylinder graph* with k rows and $L + 2r$ columns, each column being a cycle, where L and r are parameters to be set later in Definition 3.9. When $k = 2$, it is a grid which we still denote by G_{cyl}. Every vertex of G_{cyl} has degree 4 except those on the first and last columns, which have degree 3. Denote the rows by $1, \ldots, k$ and the columns by $1, \ldots, L + 2r$. We call vertices on columns $[1, r], [r + 1, r + L], [r + L + 1, L + 2r]$ the *left, middle, right part* of the graph, respectively. We will specify parameters L and r in terms of k later.

Definition 3.4 (Concrete graph compression). We let $(G_{\text{cyl}}, \equiv_V, \equiv_E)$ be a graph compression on the cylinder G_{cyl} defined as follows.

1. *Compatible vertex equivalence* $\equiv_V$. We pick factors $m_1, \ldots, m_k$ of L which are all greater than 2, called the *moduli of rows*. We define a vertex equivalence relation where, on each row i, $(i, a) \equiv_V (i, b)$ if both vertices fall in the middle part (i.e., $a, b \in [r + 1, r + L]$) and $a - b = 0 \mod m_i$.

2. *Edge equivalence* $\equiv_E$. The above $\equiv_V$ induces an edge equivalence relation on $E(G_{\text{cyl}})$ as in Definition 3.1: we order the edges incident to a vertex by the canonical choice (left, right, up, down), adjusted to a subset of size 3 for vertices on boundary columns.

Definition 3.5 (Concrete compressed Tseitin). We use $\text{Ts}(G_{\text{cyl}})$ to denote the Tseitin formula where the only odd charge is at vertex $(1, 1)$. Then the edge equivalence $\equiv_E$ induces a variable identification on $\text{Ts}(G_{\text{cyl}})$ and hence a new CNF formula, which we denote by $\tau(G_{\text{cyl}})$.

We make the following three observations about the compression.

Observation 3.6. In the edge equivalence $\equiv_E$ on $E(G_{\text{cyl}})$, a horizontal edge can be equivalent only to horizontal edges in the same row, and a vertical edge only to vertical edges between the same two adjacent rows.

Observation 3.7. We can visualize the compression as follows. Fixing a row i, we use a to represent vertex (i, a). Then each vertex on the left and right parts of G_{cyl} is a singleton vertex class, and there are m_i vertex classes in the middle part. In other words, the set of vertices in row i is partitioned into $2r + m_i$ many subsets according to $\equiv_V$: the singletons $\{1\}, \ldots, \{r\}, \{L + r + 1\}, \ldots, \{L + 2r\}$, and $(r + 1)_{\equiv_V}, \ldots, (r + m_i)_{\equiv_V}$, where $x_{\equiv_V} = \{y \in [r + 1, r + L] \mid y = x \mod m_i\}$. The horizontal edge classes are partitioned into $2(r - 1) + m_i$ subsets: a singleton for each edge within the left or within the right part,

13
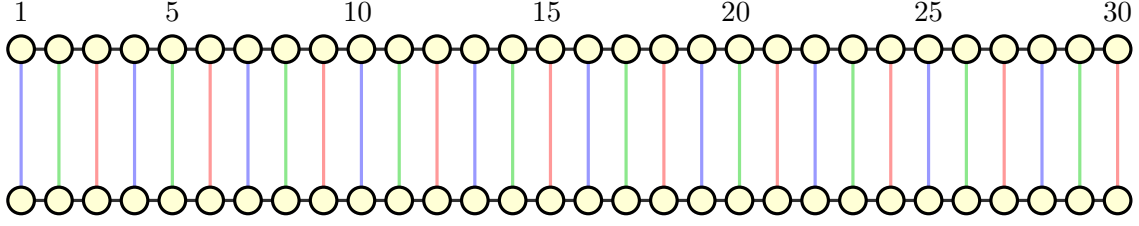
(a) Before compression.
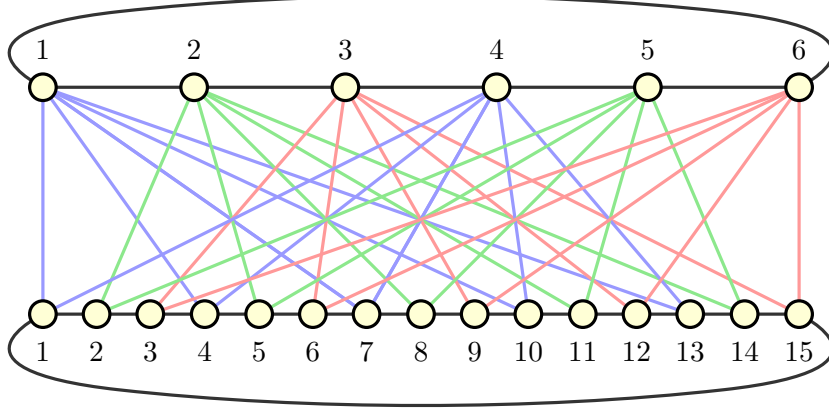
(b) After compression.

Figure 3: The compression of two rows in the middle part is depicted. The parameters are chosen as $L = P_1 \cdot P_2 \cdot P_3 = 2 \cdot 3 \cdot 5$, $c = 1$, $m_1 = P_1 \cdot P_2 = 6$, and $m_2 = P_2 \cdot P_3 = 15$. Equivalent vertical edges are drawn in the same color.

and $H_i(1), \ldots, H_i(m_i)$, where $H_i(a) = \{\{x, x+1\} \mid x \in [r, r+L], \ x = a \mod m_i\}$. Note that the two edges $\{r, r+1\}$ and $\{r+L, r+L+1\}$ that "cross parts" are equivalent and fall in $H_i(r)$ since $m_i | L$. As for the vertical edges, they are all between adjacent rows. Between rows i and $i+1$ (counted mod k), the vertical edges within the left or the right part give $2r$ singleton edge classes. In the middle part, using the Chinese remainder theorem, it is not hard to see that there are $g_i := \gcd(m_i, m_{i+1})$-many vertical edge classes $T_i(1), \ldots, T_i(g_i)$, where

$$T_i(a) = \big\{\{(i, x), (i+1, x)\} \mid x \in [r+1, r+L] \text{ and } x - a = 0 \mod g_i\big\}. \tag{3.1}$$

See Figure 3 for an illustration.

Observation 3.8. The number of vertex class is $|V(G_{\text{cyl}})/ \equiv_V| = 2kr + \sum_{i=1}^{k} m_i$, and the number of edge class is $|E(G_{\text{cyl}})/ \equiv_E| = (8r - 2)k + \sum_{i=1}^{k}(m_i + \gcd(m_i, m_{i+1}))$.

In this and the next sections, given n, k, and c, we set the parameters m_i, L, and r in G_{cyl} as follows.

Definition 3.9 (Parameters in construction). For fixed $k \geq 2$ and $c \in [1, k-1]$, let $n > 4k + 2$ be an integer such that there are k pairwise coprime numbers $P_1, \ldots, P_k$ in $[n, 2n]$. (The prime number theorem implies that having $n > 2k \ln k + C_{abs}$ suffices, where C_{abs} is an absolute constant.) For $i \in [k]$, we choose the parameters in Definition 3.4 as follows.

$$m_i := 2(k + c) \cdot P_i \cdots P_{i+c}, \tag{3.2}$$
$$L := 2(k + c) \cdot P_1 \cdots P_k, \tag{3.3}$$
$$r := k + c + 1. \tag{3.4}$$

The condition $P_1, \ldots, P_k \in [n, 2n]$ in Definition 3.9 is only for the estimates $m_i \leq k^2(2n)^{c+1}$ and $L \geq k^2 n^k$ to hold. The more structural properties we need of the parameters are summarized as follows.

14

Property 3.10 (Properties of parameters). For any subset $I \subseteq [k]$ of rows, denote $g_I := \gcd(m_i \mid i \in I)$. Then the following holds for the parameters in Definition 3.9.

(P1) $g_{[k]} \geq 2(k + c)$;

(P2) For all $i \in [k]$ and all $a, b \geq 1$ such that $a + b \leq c + 1$, $m_i \mid \text{lcm}\left(g_{[i-a,i]}, g_{[i,i+b]}\right)$. Here, $[i - a, i]$ and $[i, i + b]$ are cyclic intervals modulo k;

(P3) For all $I \subseteq [k]$ with $|I| \geq k - c$, it holds that $\text{lcm}(m_i \mid i \in I) = L$;

(P4) $r > k + c$.

Let us see that the parameters indeed satisfy the above. It is clear for (P1) and (P4). For (P2), by (3.2) we have that $A := g_{[i-a,i]}/(2(k+c))$ is divisible by $P_i, \ldots, P_{i-a+c}$ and $B := g_{[i,i+b]}/(2(k+c))$ is divisible by $P_{i+b}, \ldots, P_{i+c}$. Since $i + b \leq i - a + c + 1$, we get that $\text{lcm}(A, B)$ is divisible by $P_i, \ldots, P_{i+c}$. For (P3), note that every P_i appears (as a factor) in $c + 1$ rows by (3.2), so if $|I| \geq k - c$, then P_i is in m_j for some $j \in I$ and hence $\text{lcm}(m_i \mid i \in I) = P_1 \cdots P_k = L$.

Finally, by Remark 3.3 and Observation 3.8, the compressed Tseitin formula $\tau(G_{\text{cyl}})$ has width 4, and N variables and $\Theta(N)$ clauses, for $2k^2 n^{c+1} \leq N \leq 40k^2(2n)^{c+1}$.

3.2   The Compressed Game

We are ready to define the compressed vertex Cop-Robber game.

Definition 3.11 (Compressed Cop-Robber game). Given a graph compression $(G, \equiv_V, \equiv_E)$, the *compressed (vertex) k-Cop-Robber game* proceeds on G, where the Cops and the Robber stay on vertices and are always visible to each other. Initially, the Robber is at a vertex, and all k Cops are lifted from the graph. In a round, the following happens in turn:

(G1) If there is no lifted Cop, choose and lift one. Then, a lifted cop signals a vertex v to the Robber;

(G2) The Robber does a *compressible move* from his current vertex w_1 to some w_2, which means he provides an edge set $M \subseteq E$ such that

    (a) (M is closed under $\equiv_E$.) Whenever $e \equiv_E e'$, $e \in M \Leftrightarrow e' \in M$;

    (b) (No vertex class is occupied.) If we denote the set of vertices occupied by Cops (the *Cop position*) after (G1) by $\mathbb{C} \subseteq V$, then $\mathbb{C} \cap V(M) = \emptyset$;

    (c) (Parity flip.) For all $u \in V(M)$, $\deg_M(u)$ is odd if and only if $u \equiv_V w_1$ or $u \equiv_V w_2$. (Note that this implies $w_1 \neq w_2$.)

(G3) A lifted Cop lands at the signaled vertex v.

The game ends when a Cop is at a vertex in the Robber's vertex class.

Remark 3.12. By definition of a compressible move M, if $M \subseteq E$ is closed under $\equiv_E$, then so is $V(M)$ under $\equiv_V$, so condition (G2)b is the same as requiring that $\mathbb{C}_{\equiv_V} \cap V(M)_{\equiv_V} = \emptyset$. In the trivial case where all vertices form singleton equivalence classes, and hence so do all edges, a compressible move from w_1 to w_2 is a path connecting w_1 and w_2 plus some edge disjoint cycles that can be removed.

3.3   From Narrow Proofs to Cop Strategies

Before diving into the compressed game, let us show two facts about the refutation width of Tseitin formulas and about simulating refutations by a Cop strategy. The arguments are somewhat straightforward.

Lemma 3.13 (Small-width refutation). *The formula $\tau(G_{\text{cyl}})$ has a resolution refutation with width $k+3$ and size $O((L + r)2^k k)$.*

Proof. We view a clause as its minimum falsifying assignment and construct a proof as a top-down DAG, where we query the value of a variable at each node until arriving at an axiom. We first query the values of all right-going edges of column 1. Denote this edge set by E_1. Since E_1 separates the graph (i.e., its removal disconnects the graph into two components), at each of the 2^k branches, the restricted formula contains a subset of clauses that form a Tseitin contradiction on either the induced subgraph on column 1 or the sub-cylinder after column 1. At any of the 2^{k-1} nodes where the contradiction is on the 1st column, we continue to query the up, down edges of vertex $(1, 1)$, to arrive at either an axiom at this vertex (clause width 3) or a node in the DAG where we forget the right edge of $(1, 1)$ (clause width $k + 1$). Then we continue to do so for vertices $(1, 2), (1, 3), \ldots$ until we cyclically get back from row k to row 1, where we end up at an axiom at vertex $(1, 1)$. Note this part of the proof has width $k + 1$ and size $O(2^k k)$.

At any of the other 2^{k-1} nodes, the Tseitin instance is on the induced subgraph from the second to the last column. We query the up, down, right edges of $(2, 1)$ to arrive at either an axiom about this vertex (clause width 4), or, by forgetting its left edge, a node where the assignment has domain $E_{1,1} := E_1 \setminus \{\text{left edge of } (2, 1)\} \cup \{\text{up, right, down edges of } (1, 2)\}$ (clause width $k + 2$). Then we query the right edge of $(2, 2)$ to arrive at another node (clause width $k + 3$), and then query the down edge of $(2, 2)$ to arrive at either an axiom about vertex $(2, 2)$ or, by forgetting the left and up edges, a node where the assignment is on $H_{1,1} \setminus \{\text{up, left edges of } (2, 2)\} \cup \{\text{right, down edges of } (2, 2)\}$ (clause width $k + 2$). We continue in this fashion for vertices $(2, 3), (2, 4), \ldots$ while staying in width $k + 3$. At last, we forget 3 edges (up, left, down) of vertex $(2, k)$ to arrive at a node whose assignment is on E_2, the set of all right edges from column 2. In this process, we maintained that the edges mentioned by a non-axiom clause separate the graph, so at any node whose assignment is on E_2, the restricted formula is a Tseitin contradiction on the sub-cylinder after column 2. This part of proof has width $k + 3$ and size $O(2^k k)$.

The rest of the proof is clear: we repeat the above paragraph to $E_3, E_4, \ldots, E_{L+2r}$ until the last column, and on the last column the query processwhere the proof is symmetric to the case one we did on the first column 1. $\qquad\square$

We will only consider Tseitin formulas that are *nice with respect to a graph compression* $(G, \equiv_V, \equiv_E)$, which means that the parities at v, v' are the same if $v \equiv_V v'$, and that there is a total assignment $\alpha_{\mathrm{Ts}} : E(G)/_{\equiv_E} \to \mathbb{F}_2$ that satisfies all axioms except for at one vertex class. Note that the instance $\tau(G_{\mathrm{cyl}})$ is nice with respect to $(G_{\mathrm{cyl}}, \equiv_V, \equiv_E)$ (Definition 3.4 and Definition 3.5), with the total assignment being all-zero and the unique vertex class with odd charge being $(1, 1)$, which is a singleton class.

Lemma 3.14 (Cops simulate refutation). *The following holds for any graph compression $(G, \equiv_V, \equiv_E)$ and nice $\mathrm{Ts}(G)$ with associated assignment α_{Ts}. If there is a width-w and depth-d resolution refutation of the compressed $\mathrm{Ts}(G)$, then for the compressed $(w + 1)$-Cop-Robber game where the Robber starts at a vertex in the unique falsified class of α_{Ts}, the Cops can win in $d + 1$ rounds.*

Proof. For ease of notation, in this proof, we write $[e]$ for the $\equiv_E$-class of $e \in E(G)$, and $[v]$ for the $\equiv_V$-class of $v \in V(G)$. Given a refutation π of the compressed $\mathrm{Ts}(G)$, the Cops travel down the proof DAG π from the empty clause $\bot$. At each clause D, a new game round begins. Assume that the Robber is at v_D and that the Cop position is $\mathbb{C}_D \subseteq V(G)$. The Cops keep the following invariants.

1. $|\mathbb{C}_D| \le w$;
2. Each variable $x_{[e]}$ in D is associated to a vertex $v \in \mathbb{C}_D$ that is incident to some edge in class $[e]$;
3. They have a total assignment α_D on $\{x_{[e]} \mid e \in E(G)\}$ that falsifies only the parity at v_D, and moreover, $\alpha_D(D) = 0$.

Initially, all Cops are lifted and $\alpha_\bot = \alpha_{\mathrm{Ts}}$, so the invariants hold. Assume that the process arrives at clause D keeping the invariants and the two precedent clauses are D_1 and D_2, and denote the resolved variable at D by $x_{[e_D]}$. Since $|\mathbb{C}_D| \le w$ and there are $w + 1$ Cops in total, there is at least one lifted Cop. A lifted Cop then signals a vertex v that is adjacent to some edge in class $[e_D]$. The Robber does a compressible move M from v_D to some v'_D, and a lifted Cop lands at v. We choose the total assignment α' to be one such that, for every edge e, it flips the value $\alpha_D([e])$ if and only if $e \in M$. This is well-defined

since M is a compressible move and by (G2)a. The simulation now proceeds to the precedent clause where α' falsifies the literal over $x_{[e_D]}$, say clause D_1.

Let $\alpha_{D_1} := \alpha'$, and let us see that invariant **(3)** holds. First, α' falsifies only the parity constraint at $[v'_D]$ where v'_D is the Robber's current vertex. This is because α_D only falsifies the parity at $[v_D]$ and M flips only the parities at $[v_D]$ and $[v'_D]$ by (G2)c. Second, $\alpha'(D_1) = 0$. This is because α' falsifies the literal over $x_{[e_D]}$ in D_1, and $\alpha = \alpha'$ on the rest of the literals in D_1 since their underlying variables all have some associated vertex in $\mathbb{C}_D$ by the inductive hypothesis on **(2)**, so no edge in these classes can be used in the compressible move by (G2)b.

Now we associate the variable $x_{[e_D]}$ to the newly landed Cop, and lift a Cop that is not associated to any variable in $Vars(D_1)$, if there is one. Then there must be least one Cop in the helicopter since at most $|D_1| \leq w$ many Cops are associated to some variable in $Vars(D_1)$. So, if we let $\mathbb{C}_{D_1}$ denote the remaining Cops' positions, then $|\mathbb{C}_{D_1}| \leq w$, and thus the invariants **(1)** and **(2)** hold. This completes the induction, where a resolution step gives a game round.

Finally, when we reach an axiom clause, the clause is by definition expanded from the parity constraint at a vertex class. By **(3)** the total assignment we keep falsifies this axiom and thus the parity at this vertex class, so by the same invariant, the Robber's position u is in this class. The variables mentioned in this axiom are precisely the edge classes of all incident edges to u, so by **(2)**, either a Cop is occupying some $u' \equiv_V u$ and they win, or every neighbor of the Robber has a Cop in its class, in which case a lifted Cop can be sent to u in the next round while the Robber has no compressible move, and the Cops win. $\square$

4 Lower Bound Proof for the Cop-Robber Game

In this section, we prove the lower bound for the compressed game (Theorem 2.1), stated formally below. We use the construction $(G_{\text{cyl}}, \equiv_V, \equiv_E)$ from the previous section, including the definition of the parameters L and r from Definition 3.9.

Theorem 4.1 (Cop-Robber, formal). *For any $k \geq 2$ and $c \leq k - 1$, we have that $k + 1$ Cops can win the compressed game on $(G_{\text{cyl}}, \equiv_V, \equiv_E)$, but as long as there are at most $k + c$ Cops, the Robber can survive $(L - 2r)/(8(k + c))$ rounds.*

From this theorem, together with the relation between the Cops-Robber game and resolution width and depth (Lemma 3.14) and the upper bound for resolution (Lemma 3.13), we immediately get the supercritical width-depth trade-off stated in Theorem 2.3.

In Section 4.1, we prepare the concepts to be used in the proof. In Section 4.2, we describe the idea of the Robber's strategy and prove a key lemma. We provide the Robber's strategy to survive many rounds against $k + c$ Cops in Section 4.3, proving Theorem 4.1. Henceforth, we fix $k \geq 2$ and $c \in [1, k - 1]$, and consider the graph compression $(G_{\text{cyl}}, \equiv_V, \equiv_E)$ with parameters as in Definition 3.9. We write $V := V(G_{\text{cyl}})$, and $E := E(G_{\text{cyl}})$.

4.1 Preparations

We begin with the following special kind of compressible moves of the Robber.

Definition 4.2 (I-periodic path). Given a nonempty row set $I \subseteq [k]$ and columns a and b in $[1, 2r + L]$ where $a < b$, let $g_I := \gcd(m_i \mid i \in I)$. An *$I$-periodic path* between a and b is a path in the induced subgraph $G_{\text{cyl}}|_{I \times [a,b]}$ that can be obtained as follows. Take a simple path in $G_{\text{cyl}}|_{I \times [a, a+g_I]}$ from column a to column $a + g_I$ such that the path starts and ends in the same row, and its restrictions to column a and to column $a + g_I$, when viewed as subsets of the cycle on $[k]$, share no edges. Then extend this path periodically on $G_{\text{cyl}}|_{I \times [a+g_I, a+2g_I]}$ until hitting column b for the first time.

Using I-periodic paths, we obtain the following Robber moves, generalizing those in [GLNS23] which correspond to the case where $|I| = 2$.
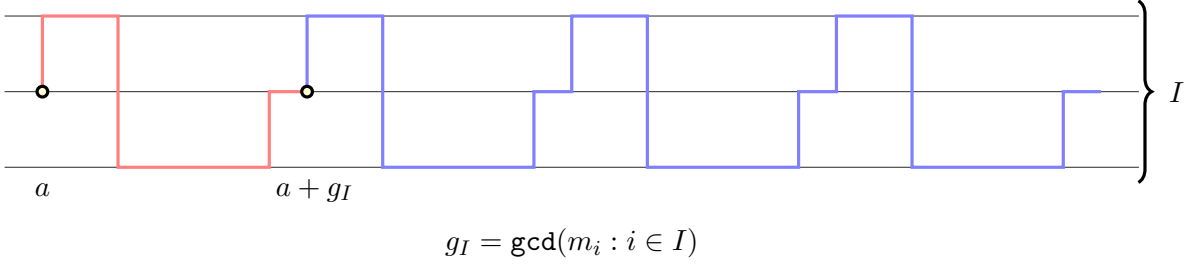
$$g_I = \texttt{gcd}(m_i : i \in I)$$

Figure 4: A special move P starting in column a is shown as the segmented path, where I is the underlying row set.

Remark 4.3 (Periodic paths give special compressible moves). Suppose W is the set of vertices occupied by Cops in the compressed game (Definition 3.11). If P is an I-periodic path ($I \neq \emptyset$) from vertex v_1 on column 1 to vertex v_2 on column $2r + L$ respectively and it avoids W, then P is a compressible move. We will call such path a *special move* of the Robber (cf. Figure 4), if he is at v_1.

To see why it is a compressible move, note that P is a complete g_I-periodic (horizontally) repetition of some edges in $I \times [1, 1 + g_I]$. Using the fact that g_I divides every row modulus in I, by Observation 3.7 we have that the edges in P are closed under $\equiv_E$, thus Condition (G2)a holds. Condition (G2)b holds by assumption. Condition (G2)c holds since P is a path connecting v_1 and v_2 which are singleton classes in $\equiv_V$.

In fact, more is true for the above special move P. For any two columns a and b in the left and right part of G_{cyl}, respectively, if v_a is the *first* intersection of P with column a and v_b is the *last* intersection of P with column b, then the subpath of P from v_a to v_b is also a compressible move, which we call the (a, b)-*truncation of* P. To see why, note that Condition (G2)b is satisfied since we took a subset of P. The parts of P before v_a and after v_b consist of singleton-class edges, so after removing them, the subpath (as an edge set) is still closed under $\equiv_E$, thus Condition (G2)a is satisfied. Finally, the subpath connects v_a and v_b which are singleton-classes in $\equiv_V$, so the parity Condition (G2)c is satisfied.

Next, we set up some notation for vertex sets and vertex separators. For $W \subseteq V$ and a row set $I \subseteq [k]$, we let $W_I := W \cap (I \times [1, 2r + L])$, which is the restriction of W on I. We let $W_{\equiv_I} := \{v \in V \mid (\exists v' \in W_I) \text{ s.t. } v, v' \text{ are on the same row and their distance is 0 mod } g_I\}$, which is the vertex set g_I-periodically generated by W_I. We call a column W-*free* if it contains no vertex in W. We call a vertex set $I \times [b, b + g_I] \subseteq I \times [1, 2r + L]$ a *good period of* $W_{\equiv_I}$ if the column b is $W_{\equiv_I}$-free.

Definition 4.4 (Vertex separators). We say a vertex set $W \subseteq V$ is a *vertex separator for* $I \times [a, b]$ if in the induced graph $G_{\text{cyl}}|_{I \times [a,b]}$ there is no path from column a to column b avoiding $W \cap (I \times [a, b])$.

We say a vertex set W is I-*separating* if in the induced graph $G_{\text{cyl}}|_{I \times [1, 2r+L]}$ there is no I-periodic path from column 1 to column $2r + L$ that avoids $W_{\equiv_I}$. We say W is a-*separating*, where $1 \leq a \leq k$, if it is I-separating for all I such that $1 \leq |I| \leq a$.

We now record a simple fact that we will use later on, and then give some basic properties of separating sets (Proposition 4.6) and vertex separators (Proposition 4.7).

Fact 4.5. *For any vertex set $R \subseteq I \times [a, b]$, where $a \leq b$, it holds that R separates columns a and b in the graph $G_{\text{cyl}}|_{I \times [a,b]}$ if and only if R separates columns 1 and $2r + L$ in $G_{\text{cyl}}|_{I \times [1, 2r+L]}$. In particular, R is a vertex separator for $I \times [a, b]$ if and only if R is a vertex separator for $I \times [1, 2r + L]$.*

Proposition 4.6. *For any $W \subseteq V$ and $\emptyset \neq I \subseteq [k]$, if $|W_I| < g_I$, then the following three statements hold.*

1. *There is at least one $W_{\equiv_I}$-free column, and $W_{\equiv_I}$-free columns appear g_I-periodically.*

2. *W is I-separating if and only if $W_{\equiv_I}$ is a vertex separator for each good period of $W_{\equiv_I}$.*

3. W is I-separating if and only if $W_{\equiv_I}$ is a vertex separator for $I \times [1, 2r + L]$.

Proof. To see Item 1, note that since $\left|W_{\equiv_I} \cap (\text{columns } [1, g_I])\right| = |W_I| < g_I$, there is a column in $[1, g_I]$ that is $W_{\equiv_I}$-free. The g_I-periodicity of $W_{\equiv_I}$-free columns is immediate.

To prove Item 2, assume that W is not I-separating. Then any I-periodic path from column 1 to column $2r + L$ avoiding $W_{\equiv_I}$ certainly implies that $W_{\equiv_I}$ is not a vertex separator for any $G_{\text{cyl}}|_{I \times [b, b+g_I]}$, as witnessed by a subpath between columns $[b, b + g_I]$. (Since any edge connects vertices in the same or adjacent columns, we can take e.g. the subpath from the last time of visiting column b to the next first time of visiting column $b + g_I$.)

For the other direction, suppose $W_{\equiv_I}$ is not a vertex separator on a good period of $W_{\equiv_I}$. By the g_I-periodicity of $W_{\equiv_I}$, this is the case for every good period of $W_{\equiv_I}$. By Item 1 a good period exists, say it is $I \times [b, b+g_I]$, then $W_{\equiv_I}$ is not a vertex separator for $I \times [b, b+g_I]$, so there is a path $P = (P(1), \dots, P(t))$ in $G_{\text{cyl}}|_{I \times [b, b+g_I]}$ from column b to $b + g_I$ avoiding $W_{\equiv_I}$. By truncating P if necessary, we can assume that $P(1)$ is the last vertex of P on column b and $P(t)$ is the first vertex of P on column $b + g_I$. Then we extend P on column $b + g_I$ to the same row as $P(1)$, which is possible since I is a segment in the row set and column $b + g_I$ is $W_{\equiv_I}$-free. Finally, this extended path can be further extended periodically in both directions to columns 1 and $2r + L$, guaranteeing that it remains a simple path. The final path witnesses that W is not I-separating.

We proceed to Item 3. If $W_{\equiv_I}$ is a vertex separator for $I \times [1, 2r + L]$, then there is no path from column 1 to column $2r + L$, so in particular there is no such path that is I-periodic and in the induced subgraph, so W is I-separating. For the other direction, assume W is I-separating. By Item 2, $W_{\equiv_I}$ is a vertex separator for every good period of itself. In particular, for any good period $I \times [b, b + g_I]$ of $W_{\equiv_I}$ (which exists by Item 1), since $W_{\equiv_I} \cap I \times [b, b + g_I] \subseteq I \times [b, b + g_I]$ is a vertex separator for $I \times [b, b + g_I]$, we get by Fact 4.5 that $W_{\equiv_I} \cap I \times [b, b + g_I]$ is a vertex separator for $I \times [1, 2r + L]$, and hence so is $W_{\equiv_I}$. $\qquad\square$

In the following, we use $\text{diam}(W)$ for the difference of the maximum and minimum indices of columns on which W is nonempty. The term *minimal* will always be used with respect to the set-inclusion relation.

Proposition 4.7. *If a set S is a minimal vertex separator for G_{cyl}, then $\text{diam}(S) \leq |S| - 1$.*

Proof. Suppose not, then there are columns $a < b$ on which S is nonempty and $b - a \geq |S|$. Thus there is a column $c \in (a, b)$ that is S-free. The set S must be a vertex separator for either $G_{\text{cyl}}|_{[k] \times [1, c]}$ or $G_{\text{cyl}}|_{[k] \times [c, 2r+L]}$, say the former. But then $S_{[k] \times [1, c]}$ is also a vertex separator for G_{cyl}, contradicting the minimality of S. $\qquad\square$

Finally, we define the concept of virtual cordons, which is key to our improved analysis over [GLNS23]. To simplify the notation, given a vertex set $W \subseteq V$ and row $i \in [k]$, we use the abbreviation W_i for $W_{\{i\}} = W \cap (\{i\} \times [1, 2r + L])$. We use $\text{row}_{\text{unique}}(W) \subseteq [k]$ to denote the set of rows where W has at most one vertex, and call them the *unique rows of W*.

Definition 4.8 (Virtual cordons[2]). Given a vertex set W, a vertex separator S for G_{cyl} is called a *virtual cordon of W* if $|S_i| \leq |W_i|$ on every row $i \in [k]$, and $S_j \subseteq (W_j)_{\equiv_V}$ for every $j \in \text{row}_{\text{unique}}(W)$.

Definition 4.9 (Critical set). We say a vertex set W is *critical* (to the Robber) if

1. W is $(c + 1)$-separating, and
2. there exists a virtual cordon of W.

Intuitively, the set of virtual cordons of W can be viewed as a kind of 'closure' of the set of vertex separators in $W_{\equiv}$ (see Figure 5). In particular, if W is the Cop position, then a virtual cordon is a vertex separator which the Cops can potentially occupy while maintaining their positions on the unique rows. We

---

[2]We used the term *semi-separators* for this definition in an earlier manuscript, which was subsequently adopted in the follow-up work [BLVS24]. We think *virtual cordons* is more descriptive.
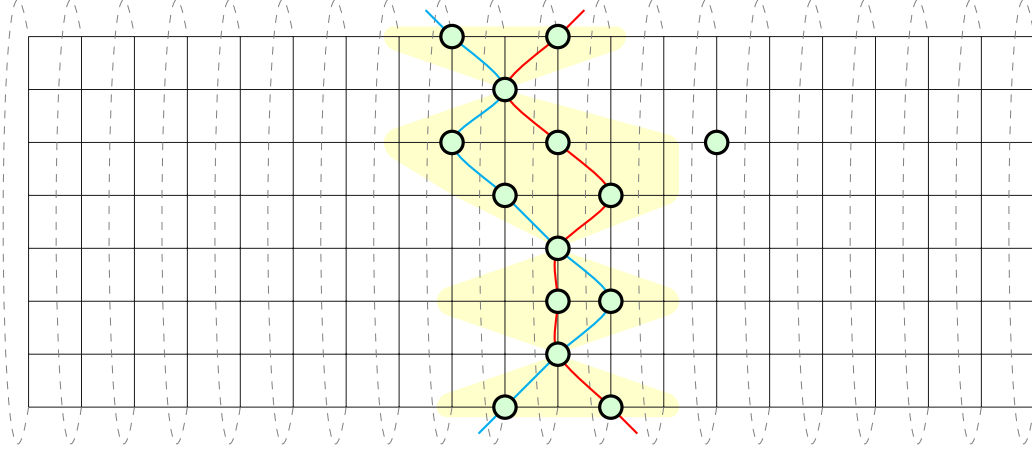
Figure 5: The green circles are vertices in $W_{\equiv}$ (i.e., they are in the equivalent classes of the Cops). The red and blue curves illustrate two minimal vertex separators contained in $W_{\equiv}$. The yellow region represents all virtual cordons associated with W.

will later design a Robber strategy against this closure, where critical Cop positions play an important role in the analysis (see Section 4.2).

Regarding the definition of critical sets, we note that both conditions in are monotone: if $W \subseteq W'$ and W is critical, then so is W'. We also remark that the $c+1$ in this definition can be replaced with any larger number while the proofs in later sections can be adapted accordingly. We pick $c+1$ for concreteness.

We end this subsection with the following property of virtual cordons.

Proposition 4.10. *Let $W \subseteq V$. Suppose that $|W| \leq k+c$ and that W is 1-separating. Then any two minimal virtual cordons of W, should they exist, coincide on* $\mathrm{row}_{\mathrm{unique}}(W)$.

Proof. Let S_1 and S_2 be two minimal virtual cordons of W. Since W is 1-separating, on each row it contains at least one vertex, meaning that

$$|\text{non-unique rows of } W| \leq (k+c) - k = c. \tag{4.1}$$

We list $\mathrm{row}_{\mathrm{unique}}(W)$ as $i_1, \ldots, i_t$, increasing in $[k]$. It holds $t \geq k - c$. For $j = 1, .., t$, let $(i_j, a_j) \in S_1$, and $(i_j, b_j) \in S_2$ denote the unique vertices on row j in S_1 and S_2, respectively.

If $c = k - 1$, then $m_1 = \ldots = m_k = L$, meaning there is no compression at all. In this case, the proposition trivially holds.

So we can assume that $c \leq k - 2$, and hence $t \geq 2$. Since minimal virtual cordons of W are minimal vertex separators for G_{cyl} as well, by their size condition and Proposition 4.7, we have that $|a_2 - a_1|, |b_2 - b_1| \leq |W| - 1 < k + c$. So, denoting by $\Delta_1 := (a_2 - a_1) - (b_2 - b_1)$, then $|\Delta_1| < 2(k+c)$. Now, since i_1 and i_2 are unique rows, $a_1 - b_1$ and $a_2 - b_2$ are multiples of m_{i_1} and m_{i_2}, respectively, and thus their difference Δ_1 is a multiple of $\gcd(m_{i_1}, m_{i_2})$. Property (P1) implies that $\gcd(m_{i_1}, m_{i_2}) \geq 2(k+c)$. Since $|\Delta_1| < 2(k+c)$ and Δ_1 is a multiple of a number greater than $2(k+c)$, it must be that $\Delta_1 = 0$.

The same argument shows that $\Delta_j := (a_{j+1} - a_j) - (b_{j+1} - b_j)$ is zero for $j = 1, \ldots, t$ (where $t+1$ is taken as 1). So $a_1 - b_1 = \ldots = a_t - b_t$ are all equal to some number Δ which is in turn a multiple of each m_{i_j} for $j \in [t]$, and hence of $M = \mathrm{lcm}(m_{i_1}, \ldots, m_{i_t})$. Now $t \geq k - c$ so by Property (P3), $M = L$ and so $L \mid \Delta$. Since $\Delta = a_1 - b_1 < 2L$, it can only be 0 or L. If $\Delta = L$ then one of a_1, b_1, say a_1, is a singleton class, a singleton class, implying that $a_1 \not\equiv_V b_1$, which contradicts the condition on virtual cordons on unique rows. So $\Delta = 0$ and thus $a_j = b_j$ for all $j = 1, \ldots, t$, proving the proposition. So $\Delta = 0$ and thus $a_j = b_j$ for all $j = 1, \ldots, t$, proving the proposition. $\square$

4.2 The Key Lemma

In the Robber strategy, we will always let the Robber stay on either the left or the right part of the graph. If the Cop position W is not critical, the Robber can either use special moves to switch sides (if W is

not $(c+1)$-separating), or move within the same part to avoid the Cops (if W has no virtual cordons and hence no vertex separators). In either case, the Cops cannot catch him. The worrying case is when W is about to become critical in a round and the Robber cannot switch sides.

Lemma 4.11 below says that this worry is unnecessary. Namely, in a round when the Cops go from having no virtual cordon to having one, the Robber has special moves to both sides of the graph.

Lemma 4.11. *Suppose $W \subseteq V$ is critical and $|W| \le k + c$. If a subset $W^- \subseteq W$ has size $|W| - 1$ and is $(c+1)$-separating, then W^- must be critical as well.*

Proof. Note the assumption implies that W is 1-separating, so as in (4.1) we have

$$|\text{non-unique rows of } W| \le c. \tag{4.2}$$

Since $W = W^- \cup \{v\}$, for some vertex v, we consider two cases:

  (A) v is on a unique row i of W^-; or
  (B) v is on a non-unique row i of W^-.

In both cases, our goal is to construct a virtual cordon of W^-.

Case (A). We may assume i is not a unique row of W, since otherwise $W_{\equiv_V} = W^-_{\equiv_V}$ and the conclusion follows. Suppose $v = (i, x)$ is the unique element of W^- on row i.

Take a *minimal* virtual cordon S of W. Let i_1 and i_2 be the two nearest unique rows of W, above and below i cyclically, which exist by (4.2) and since $c \le k - 1$. Note they are also the two nearest unique rows of W^- above and below i by the case assumption, and these two rows can coincide in the case $c = k - 1$. Define two cyclically down-going intervals $I_1 := [i_1, i]$ and $I_2 := [i, i_2]$. Then by (4.2) again we have $(i - i_1) + (i_2 - i) \le c + 1$ (hinting that we will eventually apply Property (P2)). The idea is to construct a virtual cordon S' of W^- starting from S, which we will achieve by maintaining S on rows outside of $I_1 \cup I_2$, changing S on $I_1 \cup I_2$, and then "gluing" them together.

We first define some notation. Let (i_1, a) and (i_2, b) be the two unique vertices of S in the corresponding rows. Since S is also a minimal separator for G_{cyl}, by Proposition 4.7,

$$|a - b| < k + c. \tag{4.3}$$

Consider the graph restricted on I_1. Since $|I_1| \le c + 1$ we have, by the $(c+1)$-separating assumption, that W^- is I_1-separating. By Items 1 and 2 in Proposition 4.6, there is a good period of W^- on I_1 containing (i_1, a), in which $(W^-)_{\equiv_{I_1}}$ is a vertex separator, and similarly for I_2. We fix such good periods on I_1 and I_2 below.

Pick an S_1 that is a *minimal* subset of $(W^-)_{\equiv_{I_1}}$ such that S_1 is a vertex separator in the good period. Note that $S_1 \cap (\text{row } i) \ne \emptyset$, and since there is a unique element of $(W^-)_{\equiv_{I_1}}$ on row i in that good period, say (i, y), this element must be the only element of S_1 on i. Similarly, pick a minimal subset S_2 of $(W^-)_{\equiv_{I_2}}$ so that S_2 is a vertex separator within the good period there, and let (i, z) be the unique element of S_2 on row i.

Claim 4.12. It holds that $y = z$ and, moreover, that $(i, y) \in (W^-)_{\equiv_V}$.

Proof. By Proposition 4.7 applied to S_1 and the induced subgraph on $I_1 \times (\text{good period})$, we have that

$$|a - y| < |S_1| \le |W_{I_1}|, \tag{4.4}$$

where the last inequality holds because $W_{\equiv_{I_1}}$ within a good period has size $|W_{I_1}|$. Similarly, for I_2, we have that

$$|b - z| < |S_2| \le |W_{I_2}|. \tag{4.5}$$

Now we pick any p such that $(i, p) \in (W^-)_{\equiv_V}$. The rest of the proof of the claim is similar to that of Proposition 4.10. First we show $y = z$. On the one hand, by definition of $\equiv_{I_1}$, $p - y$ is a multiple of g_{I_1}.

Similarly, $p - z$ is a multiple of g_{I_2}. So $y - z$ is a multiple of $\gcd(g_{I_1}, g_{I_2}) = g_I$, where $I := I_1 \cup I_2$. By Property (P1) of the parameters, we have $g_I \geq 2(k + c)$. On the other hand, by (4.3), (4.4), (4.5) we have

$$|y - z| \leq |y - a| + |a - b| + |b - z| \leq (|W_{I_1}| - 1) + (k + c - 1) + (|W_{I_2}| - 1) < k + c + |W| < 2(k + c). \quad (4.6)$$

Together this means $y - z = 0$, so $y = z$. Next, note that $y - x = z - x$ is a multiple of g_{I_1} and g_{I_2} and thus of $g := \operatorname{lcm}(g_{I_1}, g_{I_2})$, so by Property (P2) g is a multiple of m_i. Therefore, $m_i | y - x$, meaning that $(i, y) \in (W^-)_{\equiv_V}$. The claim is proved. $\qquad \square$

Let us now also consider S_3 which is S restricted to the rows $I_3 := [i_2, i_1]$ (a cyclic interval mod k). By Fact 4.5, S_1, S_2 and S_3 are vertex separators for $G_{\mathrm{cyl}}|_{I_1}$, $G_{\mathrm{cyl}}|_{I_2}$ and $G_{\mathrm{cyl}}|_{I_3}$, respectively. Note that S_1, S_2 and S_3 pairwise intersect at a single vertex. Our goal is to show that their union S' is a vertex separator for G_{cyl}.

Claim 4.13 (Gluing vertex separators). Let $J_1 = [j_1, j_2]$ and $J_2 = [j_2, j_3]$ be cyclical intervals (modulo k) of rows, where j_1 and j_3 could be the same. If T_1 is a vertex separator for $G_{\mathrm{cyl}}|_{J_1}$ and T_2 is a vertex separator for $G_{\mathrm{cyl}}|_{J_2}$ such that j_1, j_2 and j_3 are among the unique rows of both T_1 and T_2, and T_1 and T_2 share their unique vertex on row j for all $j \in J_1 \cap J_2$, then $T_1 \cup T_2$ is a vertex separator for $G_{\mathrm{cyl}}|_{J_1 \cup J_2}$.

Once the claim is proved, we can use it to glue S_1 and S_2 and then to glue $S_1 \cup S_2$ and S_3 to obtain a virtual cordon of W^-, which will complete the proof for Case (A).

Proof of Claim 4.13. We prove it for the case $j_1 = j_3$; the case $j_1 \neq j_3$ is similar (and simpler). Pick any path $P = (P(1), P(2), \ldots, P(m))$ that starts from the first column in $G_{\mathrm{cyl}}|_{J_1 \cup J_2}$ and avoids $T_1 \cup T_2$. We show that P cannot reach the last column. Suppose (j_1, x_1) and (j_2, x_2) are the unique vertices in both T_1 and T_2 on the corresponding rows. Let $t_1, \ldots, t_K \in [m]$ be the times, in increasing order, where P hits rows j_1 or j_2. There is at least one such t_i since if P never hits j_1 nor j_2 then it stays in either $G_{\mathrm{cyl}}|_{J_1}$, or $G_{\mathrm{cyl}}|_{J_2}$, depending of the starting row, and cannot reach the last column since either T_1 or T_2 is a vertex separator there. We prove by induction on ℓ that if $P(t_\ell)$ is on row j_1 then it is to the left of (j_1, x_1), and similarly if it is on row j_2 it is to the left of (j_2, x_2).

Base case ($j = 1$): From $P(1)$ to $P(t_1)$ the path stays within $G_{\mathrm{cyl}}|_{J_1}$ or $G_{\mathrm{cyl}}|_{J_2}$, without loss of generality we assume the former and $P(t_1)$ is on row j_1. If $P(t_1)$ is to the right of x_1, then we can extend it on row j_1 straight to the last column. This extended path is in $G_{\mathrm{cyl}}|_{J_1}$, avoids T_1 (since T_1 on row j_1 is the singleton $\{(j_1, x_2)\}$), and reaches from left to right, contradicting the assumption that T_1 is a vertex separator for $G_{\mathrm{cyl}}|_{J_1}$.

From t_ℓ to $t_{\ell+1}$: the inductive hypothesis is that $P(t_\ell)$ is to the left of (j_1, x_1) or (j_2, x_2) depending on the row it is in. Assume without loss of generality it is j_1. Then again the path from $P(t_\ell)$ to $P(t_{\ell+1})$ falls within one of $G_{\mathrm{cyl}}|_{J_1}$ and $G_{\mathrm{cyl}}|_{J_2}$. If $P(t_{\ell+1})$ is to the right of the unique vertex on the same row, then consider a new path from (j_1, t) straight right to P_{t_ℓ} then following P to $P(t_{\ell+1})$ and then straight right to column $2r + L$. This path falls in one of $G_{\mathrm{cyl}}|_{J_1}$ and $G_{\mathrm{cyl}}|_{J_2}$ (since P from $P(t_\ell)$ to $P(t_{\ell+1})$ does) and it avoids T_1 and T_2, a contradiction. This completes the induction step.

Finally, for $P(t_K)$ to $P(m)$ the argument is symmetric to that for $P(1)$ to $P(t_1)$. This shows that $P(m)$ cannot be in the last column and the claim follows. $\qquad \square$

Case (B). This case is similar, and in fact simpler. Recall that in this case, there is a non-unique row i of W^- on which the vertex $v \in W \setminus W^-$ lies. The only reason that the virtual cordon S of W is not one of W^- is that $|S_i| > |(W^-)_i|$, and we change S as follows.

As in the previous case, the two nearest unique rows i_1 and i_2 of W^- above and below i (cyclically) exist. We denote $[i_1, i_2]$ by I. From the case assumption, W contains at least 3 vertices in row i, so the number of non-unique rows in W is at most $c - 1$. Thus, I contains at most $c + 1$ rows. Then since W^- is I-separating, we know there is a minimal vertex separator S_1 for $G_{\mathrm{cyl}}|_I$ such that on each row $j \in I$, $|(S_1)_j| \leq |W_j^-|$, and that S_1 has the same vertex as S on row i_1. By the minimality of S (as a vertex separator for G_{cyl}) and S_1 (as a vertex separator for $G_{\mathrm{cyl}}|_I$) and Proposition 4.7, both $\operatorname{diam}(S)$ an

$\operatorname{diam}(S_1)$ are at most $k + c$, so the vertex of S_1 on row i_2 and that of S have horizontal distance at most $2(k + c)$, which is smaller than g_I by (P1). But they both belong to $(W_{i_2})_{\equiv_I}$ where $\left|W_{i_2}\right| = 1$, so they coincide. Now we can apply Claim 4.13 to glue together S_1 and $S_{(I)^c}$ to form a virtual cordon of W^-.

In both cases, we have constructed a virtual cordon of W^-, so Lemma 4.11 is proved. $\qquad\square$

4.3   Proof of Cop-Robber Lower Bound

We now prove Theorem 4.1, that is, we give a Robber strategy to survive at least $L/4(c + k)$ rounds in the compressed game defined with respect to $(G_{\mathrm{cyl}}, \equiv_V, \equiv_E)$, where we choose the moduli and parameters in the construction according to Definition 3.9. Since G_{cyl} is connected, it does not matter where the Robber initially is; we assume without loss of generality that he starts at vertex $(1, 1)$, a singleton class.

At a high level, the idea is for the Robber to monitor all virtual cordons corresponding to the Cop position and consistently avoid them. To show this is possible, we will use Lemma 4.11 and the properties of separators shown in Section 4.1.

For the sake of clarity, let us denote the evolution of the Cop position over rounds by

$$(\emptyset = \mathbb{C}_1^- \to \mathbb{C}_1^+) \to \dots \to (\mathbb{C}_t^- \to \mathbb{C}_t^+) \to (\mathbb{C}_{t+1}^- \to \mathbb{C}_{t+1}^+) \to \cdots \tag{4.7}$$

That is, $\mathbb{C}_t^-$ is the Cop position in round t after Step (G1), and $\mathbb{C}_t^+$ after Step (G3). Note that $\mathbb{C}_t^-, \mathbb{C}_{t+1}^- \subseteq \mathbb{C}_t^+$, $|\mathbb{C}_t^+ \setminus \mathbb{C}_t^-| = 1$, and when the Robber moves, he knows both $\mathbb{C}_t^-$ and $\mathbb{C}_t^+$.

Invariants of the Robber strategy. In each game round, we will keep the following invariants.

(I1) The Robber is on a Cop-free column on either the left or the right part of G_{cyl};

(I2) If $\mathbb{C}_t^-$ is critical, then any minimal virtual cordon S of $\mathbb{C}_t^-$ has *horizontal distance* at least $L/2 - 4(k + c) \cdot t$ to the part (left or right) of the graph where the Robber is in.

Here, the horizontal distance between $A, B \subseteq V$, denoted by $d_h(A, B)$, is the minimum distance between columns on which A is non-empty and columns on which B is non-empty.

Proof of Theorem 4.1. To see that $k + 1$ Cops can win the game, the Cops just play as if the game is uncompressed: they first occupy the middle column of G_{cyl}, then keep moving towards the next full column in the Robber's direction. It is easy to check that $k + 1$ Cops can do this in k rounds while always maintaining that their positions form a vertex separator for $[k] \times [1, 2r + L]$.

We are left to prove the round lower bounds against $k + c$ Cops. It suffices to show that the Robber can maintain Invariants (I1) and (I2) in the first $(L - 2r)/(8(k + c))$ many rounds (since (I1) implies that the Robber is not caught), and we prove this by induction on $t < (L - 2r)/(8(k + c))$. The base case $t = 1$ is trivial since $\mathbb{C}_1^- = \emptyset$, and so (I2) vacuously holds, and the Robber is at $(1, 1)$ so (I1) holds. For the inductive step, suppose the invariants hold for $\mathbb{C}_t^-$. At the beginning of round t, assume without loss of generality the Robber is at vertex v_t in column a_t on the left part of the graph. Since both the left and right parts of the graph have $r > k + c \geq |\mathbb{C}_t^+|$ columns by (P4), each part contains a $\mathbb{C}_t^+$-free column, say column a_l on the left and a_r on the right.

Case (1). $\mathbb{C}_t^+$ is not critical. Since $\mathbb{C}_{t+1}^- \subseteq \mathbb{C}_t^+$ and the property of being critical is monotone, (I2) holds vacuously for $\mathbb{C}_{t+1}^-$, so we only need to maintain (I1).

The Robber will move to column a_l during Step (G2) of round t. Note that if $a_l = a_t$, then the Robber just moves up one row (mod k). So we consider the case where $a_l \neq a_t$. First assume that $\mathbb{C}_t^+$ violates Condition 1 of critical sets. Then we have an I-periodic path for some row set I, say going from left to right, from column 1 to column $L + 2r$. If $a_t < a_l$, then we can truncate this path to be between the first time of meeting column a_t and the first time of meeting column a_l; the case for $a_t > a_l$ is similar. Otherwise, $\mathbb{C}_t^+$ must violate Condition 2 of critical sets. In particular, $(\mathbb{C}_t^+)_{\equiv_V}$ (whose restriction to the left part is the same as that of $\mathbb{C}_t^+$) is not a vertex separator for $G_{\mathrm{cyl}}|_{[k] \times [a_t, a_l]}$. Hence, there is a path between the two columns avoiding $(\mathbb{C}_t^+)_{\equiv_V}$.

Since both a_t and a_l are $\mathbb{C}_t^-$-free columns, the Robber can move vertically to the appropriate row and take the path to column a_l, which is a compressible move and (I1) holds for $\mathbb{C}_{t+1}^-$.

Case (2). $\mathbb{C}_t^+$ *is critical but* $\mathbb{C}_t^-$ *is not.* By Lemma 4.11, $\mathbb{C}_t^-$ is not $(c+1)$-separating. So there is a compressible move (actually a path) P'' from v_t to a vertex in column a_r, for example by using an (a_t, a_r)-truncation of some I-periodic path as in Remark 4.3, plus a suitable modification on the starting vertical subpath within column a_t. There is another compressible move P', again a path, from v_t to a_l by truncating the same I-periodic path as in *Case (1)*. To decide which of the two moves to use, we first estimate the horizontal span of all minimal virtual cordons of $\mathbb{C}_t^+$. For any two minimal virtual cordons S and S', since $\left|\mathrm{row}_{\mathrm{unique}}(\mathbb{C}_t^+)\right| > 0$, we have for all $v \in S$ and $v' \in S'$ that

$$d_h(v, v') \overset{\text{Prop. 4.10}}{\leq} \mathrm{diam}(S) + \mathrm{diam}(S') \overset{\text{Prop. 4.7}}{<} 2(k+c) - 1\,. \tag{4.8}$$

In particular, the column indices of the vertices over all minimal virtual cordons of $\mathbb{C}_t^+$ can be contained in an interval $H = [h_1, h_2]$ of length $2(k+c)-1$, so $\left|h_1 - a_l\right|$ or $\left|h_2 - a_r\right|$ is at least $L/2 - (k+c)$. The Robber then takes compressible move P' or P'' so that he stays horizontally at least $L/2 - (k+c) > L/2 - 4(k+c) \cdot t$ far away from any minimal virtual cordon of $\mathbb{C}_t^+$. Both invariants hold for $\mathbb{C}_{t+1}^-$ since $\mathbb{C}_{t+1}^- \subseteq \mathbb{C}_t^+$.

Case (3). $\mathbb{C}_t^-$ *is critical.* In this case, $\mathbb{C}_t^+$ is also critical, and we assume $a_t \leq a_l$ (the case for $a_t > a_l$ is similar). Note that $\mathbb{C}_t^+$ cannot be a vertex separator for $[k] \times [a_t, a_l]$, since otherwise the horizontal distance between the Robber and a minimal virtual cordon will be smaller than r, contradicting the inductive hypothesis on (I2) as $t < (L - 2r)/(8(k+c))$. So the Robber can move from v_t to column a_l via a path within the left part and avoiding $\mathbb{C}_t^-$, which is a compressible move against $(\mathbb{C}_t^-)_{\equiv V}$, and Invariant (I1) will hold for $\mathbb{C}_{t+1}^-$.

As for (I2), note that $\mathrm{row}_{\mathrm{unique}}(\mathbb{C}_t^-) \cap \mathrm{row}_{\mathrm{unique}}(\mathbb{C}_t^+) = \mathrm{row}_{\mathrm{unique}}(\mathbb{C}_t^+)$ and that this set must be nonempty. Denote the set of minimal virtual cordons of $\mathbb{C}_t^-$ by α^-, and that of $\mathbb{C}_t^+$ by α^+. Fix any $S \in \alpha^-$. We have that S has only one vertex, which we denote by w_i, on each row $i \in \mathrm{row}_{\mathrm{unique}}(\mathbb{C}_t^+)$ by the definition of virtual cordons. By monotonicity, S is also a virtual cordon of $\mathbb{C}_t^+$, so there is a minimal $S_1 \in \alpha^+$ contained in S and, moreover, such that $(i, w_i) \in S_1$. Now for any two vertices w and w' in some members of α^- and α^+, respectively, similar to (4.8) we have that both $d_h(w, w_i)$ and $d_h(w', w_i)$ are smaller than $2(k+c) - 1$. Therefore, since the Robber stays on the same part of the graph, the horizontal distance between this part and anything in α^+ decreases by at most $\max_{w,w'}\{|d_h(w, w')|\} < 4(k+c)$ compared to α^-, so Invariant (I2) holds.

This completes the proof of Theorem 4.1. □

5 Lifting Theorem for Treelike Resolution

The aim of this section is to prove the lifting theorem for treelike resolution, restated below for convenience.

Theorem 2.4 (Restated) (Lifting for treelike resolution). *Let F be a CNF formula and let $m \geq 2$. If there is a width-w, size-s treelike resolution refutation for $\mathrm{XOR}_m(F)$, then there is a width-$\left(\frac{w}{m-1}\right)$, depth-$\log s$ resolution refutation of F.*

We reformulate this lifting theorem for the equivalent model of *decision trees*, in which the proof becomes more intuitive. Let us start by introducing some notation. Let $\mathcal{S} \subseteq \{0,1\}^n \times \mathcal{O}$ be a total search problem. A *partial assignment* to the input variables of $\mathcal{S}$ is a function $\rho : [n] \to \{0, 1, \star\}^n$ mapping the variables to 0, 1 or leaving them unassigned, which corresponds to mapping them to $\star$. We define the *fixed indices of ρ* to be $\mathrm{fix}(\rho) = \{i \in [n] : \rho(i) \neq \star\}$ and the *width* of ρ to be $|\mathrm{fix}(\rho)|$. Given two partial assignment $\rho, \rho' : [n] \to \{0, 1, \star\}^n$, we say ρ' *extends* ρ if $\rho(i) \in \{\rho'(i), \star\}$ for all $i \in [n]$. Similarly, we say $x \in \{0,1\}^n$ extends $\rho : [n] \to \{0, 1, \star\}^n$, if $\rho(i) \in \{x_i, \star\}$ for all $i \in [n]$.

We use the following definition of decision tree.

Definition 5.1 (Decision DAG and decision tree). A decision DAG solving $\mathcal{S} \subseteq \{0,1\}^n \times \mathcal{O}$ is a rooted DAG where each node v is labelled with a partial assignment ρ_v such that the following hold:

1. *Root.* The root r is labelled with the constant-$\star$ function, that is, $\rho_r(i) = \star$ for all $i \in [n]$.

2. *Non-leaf.* If v is a non-leaf node then it has two children, the 0-child and the 1-child, and it is labelled with some index $\hat{\imath}$ such that $\rho_v(\hat{\imath}) = \star$ and, for $b \in \{0,1\}$, the partial assignment of its b-child v' satisfies $\rho_{v'}(\hat{\imath}) \in \{b, \star\}$ and $\rho_{v'}(i) \in \{\rho_v(i), \star\}$ for $i \neq \hat{\imath}$.

3. *Leaf.* If v is a leaf then it is labelled with an $o \in \mathcal{O}$ such that $(x, o) \in \mathcal{S}$ for every $x \in \{0,1\}^n$ that extends ρ_v.

The *size* of a decision DAG is the number of nodes it has, the *depth* is the length of the longest root-to-leaf path in the DAG, and the *width* is the maximum over v of the width of any ρ_v.

A decision tree is a decision DAG where the underlying DAG is a tree.

Recall that $\mathsf{Search}(F) \subseteq \{0,1\}^n \times [m]$ is the search problem defined as $(x, i) \in \mathsf{Search}(F) \iff C_i(x) = 0$. The following folklore lemma relates resolution refutations of F to decision DAGs solving $\mathsf{Search}(F)$.

Lemma 5.2 (Folklore). *Let F be an unsatisfiable CNF formula. There is a width-w resolution refutation π of F with underlying DAG G_π if and only if there is a width-w decision DAG solving $\mathsf{Search}(F)$ with the same underlying DAG G_π.*

We can now state our lifting theorem for decision trees, from which Theorem 2.4 follows easily.

Theorem 5.3. *Let $\mathcal{S} \subseteq \{0,1\}^n \times \mathcal{O}$ be a search problem and let $m \geq 2$. If there is a width-w, size-s decision tree for $\mathcal{S} \circ \mathrm{XOR}_m^n$, then there is a width-$\left(\frac{w}{m-1} \right)$, depth-$\log s$ decision tree for $\mathcal{S}$.*

Proof. Given a treelike resolution refutation for $\mathrm{XOR}_m(F)$, we get by Lemma 5.2 a decision tree solving $\mathsf{Search}(\mathrm{XOR}_m(F))$, which we can turn into one solving $\mathsf{Search}(F) \circ \mathrm{XOR}_m^n$ by changing the labels on leaves. Namely, if a leaf is labelled a clause in the CNF expansion of $\mathrm{XOR}_m(C')$ for some clause $C' \in F$, we change the label to C'. Then we apply Theorem 5.3 to this decision tree and use Lemma 5.2. $\qquad\square$

Before diving into the proof of Theorem 5.3, we make a simple observation.

Claim 5.4. Fix $o \in \mathcal{O}$ and let ρ be a partial assignment to $\{0,1\}^{nm}$ such that for every $y \in \{0,1\}^{nm}$ which extend ρ, $(y, o) \in \mathcal{S} \circ \mathrm{XOR}_m^n$. Then for any $x \in \{0,1\}^n$ such that $x_i = \rho(i, 1) \oplus \ldots \oplus \rho(i, m)$ for all $i \in \mathsf{fix}(\rho)$, it holds that $(x, o) \in \mathcal{S}$.

Proof. Let $x \in \{0,1\}^n$ be such that $x_i = \rho(i, 1) \oplus \ldots \oplus \rho(i, m)$ for all $i \in \mathsf{fix}(\rho)$. Let y be the extension of ρ defined as follows: for each $j \in [n] \setminus \mathsf{fix}(\rho)$ set $y_{j,1}, \ldots, y_{j,m}$ so that $y_{j,1} \oplus \ldots \oplus y_{j,m} = x_j$. Then $\mathrm{XOR}_m^n(y) = x$. As $(y, o) \in \mathcal{S} \circ \mathrm{XOR}_m^n$, it follows by definition that $(x, o) = (\mathrm{XOR}_m^n(y), o) \in \mathcal{S}$. $\qquad\square$

We are now ready to prove Theorem 5.3. The intuition is that we solve $\mathcal{S}$ by simulating a decision tree T for $\mathcal{S} \circ \mathrm{XOR}_m^n$ top-down. At a node v of T that queries $y_{i,j}$, we move directly to the child with a smaller subtree size, unless the number of y-variables over x_i assigned by ρ_v reaches the threshold $m - 1$; in that case, we query x_i.

Proof of Theorem 5.3. Given a width-w, size-s decision tree T for $\mathcal{S} \circ \mathrm{XOR}_m^n$, we construct a decision tree $\widetilde{T}$ for $\mathcal{S}$ of width at most $\frac{w}{m-1}$ and depth at most $\log s$. Recall that for every node v of T we have a partial assignment $\rho_v : [n] \times [m] \to \{0, 1, \star\}^{nm}$, where for simplicity we view the partial assignments to the input variables of $\mathcal{S} \circ \mathrm{XOR}_m^n$ as maps from the domain $[n] \times [m]$ instead of $[nm]$. We also view the query labels in T as pairs $(\hat{\imath}, \hat{\jmath}) \in [n] \times [m]$. We define $\widetilde{T}$ by describing, for any sequence of query answers, a root-to-leaf path in $\widetilde{T}$ with the labels for each node in the path.

To define this root-to-leaf path, we start at the root of T and walk down to a leaf following some rules we describe below and occasionally querying input variables of $\mathcal{S}$ and creating new nodes in $\widetilde{T}$. At every

step, we are at some node v in T with a corresponding partial assignment ρ_v, and at some node $\widetilde{v}$ in $\widetilde{T}$ without any labels yet. In the beginning we are at the root of both T and $\widetilde{T}$. For each v in T that we traverse, we inductively define a partial assignment $\sigma_v : [n] \to \{0, 1, \star\}^n$ that will guide us in choosing the path in T to follow and in defining the path in $\widetilde{T}$. For the root r of T, we let σ_r be the constant-$\star$ function. Suppose we are at some non-leaf node v in T with query label $(\hat{\imath}, \hat{\jmath})$, and at some node $\widetilde{v}$ in $\widetilde{T}$. We distinguish three cases.

1. *(Halving case) If* $|\{j \mid \rho_v(\hat{\imath}, j) = \star\}| \geq 2$: In T, we move to the child of v that is the root of the smallest subtree, breaking ties arbitrarily. We do nothing in $\widetilde{T}$.

2. *(Forced case) If* $|\{j \mid \rho_v(\hat{\imath}, j) = \star\}| \leq 1$ *and* $\sigma_v(\hat{\imath}) \neq \star$: In T, we move to the b-child of v, where $b \in \{0, 1\}$ is such that $b \oplus \bigoplus_{j \neq \hat{\jmath}} \rho_v(\hat{\imath}, j) = \sigma_v(\hat{\imath})$. We do nothing in $\widetilde{T}$.

3. *(Query case) Otherwise,* $|\{j \mid \rho_v(\hat{\imath}, j) = \star\}| \leq 1$ *and* $\sigma_v(\hat{\imath}) = \star$: In $\widetilde{T}$, we label $\widetilde{v}$ with the partial assignment $\rho_{\widetilde{v}} = \sigma_v$ and with the query label $\hat{\imath}$. We then query the $\hat{\imath}$th variable. Let $\widetilde{b}$ be the result of this query. We create and move to the $\widetilde{b}$-child of $\widetilde{v}$. In T, we move to the b-child of v, where $b \in \{0, 1\}$ is such that $b \oplus \bigoplus_{j \neq \hat{\jmath}} \rho_v(\hat{\imath}, j) = \widetilde{b}$.

Let v' be the child of v we chose to move to. If v' is chosen in case 1 or 2, then

$$\sigma_{v'}(i) = \begin{cases} \sigma_v(i), & \text{if } |\{j \mid \rho_{v'}(\hat{\imath}, j) = \star\}| \leq 1; \\ \star, & \text{otherwise.} \end{cases} \tag{5.1}$$

and if v' is chosen in case 3, then

$$\sigma_{v'}(i) = \begin{cases} \widetilde{b}, & \text{if } i = \hat{\imath} \text{ and } |\{j \mid \rho_{v'}(\hat{\imath}, j) = \star\}| \leq 1; \\ \sigma_v(i), & \text{if } i \neq \hat{\imath} \text{ and } |\{j \mid \rho_{v'}(\hat{\imath}, j) = \star\}| \leq 1; \\ \star, & \text{otherwise.} \end{cases} \tag{5.2}$$

When we reach a leaf v of T labelled with $o \in \mathcal{O}$, we label the node $\widetilde{v}$ we are at in $\widetilde{T}$ with the answer o and with the partial assignment $\rho_{\widetilde{v}} = \sigma_v$, and the path ends (i.e., $\widetilde{v}$ is a leaf). Note that this process defines a root-to-leaf path in $\widetilde{T}$, with labels on all nodes in the path, and thus completes the description of $\widetilde{T}$. We now need to argue that $\widetilde{T}$ solves $\mathcal{S}$, has width at most $\frac{w}{m-1}$ and depth at most $\log s$.

To see why $\widetilde{T}$ is a decision tree solving $\mathcal{S}$, we need to show it satisfies items 1-3 in Definition 5.1. For the first two items we use the following observation.

Claim 5.5. *If at any given point we are at a node v in T and at a node $\widetilde{v}$ in $\widetilde{T}$, then σ_v extends $\rho_{\widetilde{v}}$.*

Proof. Let u be the last node in T such that we are at u in T and at $\widetilde{v}$ in $\widetilde{T}$ at the same time. By definition of $\rho_{\widetilde{v}}$ we have that $\rho_{\widetilde{v}} = \sigma_u$. This claim follows since all nodes between v and u are chosen in case 1 or 2, which implies σ_v extends σ_u. $\square$

Let r be the root of T and $\widetilde{r}$ be the root of $\widetilde{T}$. Item 1 is satisfied since, by Claim 5.5, σ_r extends $\rho_{\widetilde{r}}$, and σ_r is the constant-$\star$. Now, let v' be the child of v and $\widetilde{v}'$ be the $\widetilde{b}$-child of $\widetilde{v}$ chosen in case 3 after querying the $\hat{\imath}$th variable. First note that $\rho_{\widetilde{v}}(\hat{\imath}) = \sigma_v(\hat{\imath}) = \star$ by definition of $\rho_{\widetilde{v}}$ and since we are in case 3. Observe, moreover, that $\sigma_{v'}(\hat{\imath}) \in \{\widetilde{b}, \star\}$ and, for $i \neq \hat{\imath}$, that $\sigma_{v'}(i) \in \{\sigma_v(i), \star\} = \{\rho_{\widetilde{v}}(i), \star\}$. These two observations, together with the fact that, by Claim 5.5, $\rho_{\widetilde{v}'}(i) \in \{\sigma_{v'}(i), \star\}$ for all $i \in [n]$, implies $\widetilde{T}$ satisfies item 2.

To see that $\widetilde{T}$ satisfies item 3 in Definition 5.1, we need the following claim.

Claim 5.6. *For all nodes v we traverse in T, it holds that:*

$$\sigma_v(i) = \rho_v(i, 1) \oplus \ldots \oplus \rho_v(i, m) \qquad \forall i \in \mathsf{fix}(\rho_v). \tag{5.3}$$

Proof. We prove by induction on the distance from the root that the claim holds. It is not hard to see that the claim holds for the root r of T, since ρ_r is the constant-$\star$ function. Now, suppose we are at node v in T where (5.3) holds, and let v' be the child of v we choose to move to. Let $i \in \text{fix}(\rho_{v'})$ (if there is no such i then the claim holds trivially), which implies that $|\{j \mid \rho_{v'}(i, j) = \star\}| = 0$. Let $(\hat{i}, \hat{j})$ be the query label of v.

If $i \neq \hat{i}$, by definition of a decision tree we have that $|\{j \mid \rho_{v'}(i, j) = \star\}| \geq |\{j \mid \rho_v(i, j) = \star\}|$, which implies that $|\{j \mid \rho_v(i, j) = \star\}| = 0$ and thus $i \in \text{fix}(\rho_v)$. We can therefore conclude that

$$
\begin{aligned}
\sigma_{v'}(i) &= \sigma_v(i) && \text{by definition of } \sigma_{v'} \text{ since } i \in \text{fix}(\rho_{v'}) && (5.4)\\
&= \rho_v(i, 1) \oplus \ldots \oplus \rho_v(i, m) && \text{by the induction hypothesis} && (5.5)\\
&= \rho_{v'}(i, 1) \oplus \ldots \oplus \rho_{v'}(i, m) && \text{since } i \in \text{fix}(\rho_{v'}) \text{ and } \rho_{v'}(i, j) \in \{\rho_v(i, j), \star\}. && (5.6)
\end{aligned}
$$

Similarly, if $i = \hat{i}$, by definition of a decision tree we have that $|\{j \mid \rho_{v'}(i, j) = \star\}| \geq |\{j \mid \rho_v(i, j) = \star\}| - 1$, which implies that $|\{j \mid \rho_v(i, j) = \star\}| \leq 1$. This means we are either in case 2 or 3. Let b be such that v' is the b-child of v. If we are in case 2, let $\widetilde{b} = \sigma_v(i)$. We can conclude that

$$
\begin{aligned}
\sigma_{v'}(i) &= \widetilde{b} && \text{by definition of } \sigma_{v'} \text{ since } i \in \text{fix}(\rho_{v'}) && (5.7)\\
&= b \oplus \bigoplus_{j \neq \hat{j}} \rho_v(i, j) && \text{by choice of } v' && (5.8)\\
&= \rho_{v'}(i, 1) \oplus \ldots \oplus \rho_{v'}(i, m) \,, && && (5.9)
\end{aligned}
$$

where for the last equality we use that since $\rho_{v'}(i, j) \neq \star$ for all $j \in [m]$, it must be the case that $\rho_{v'}(i, \hat{j}) = b$ and for all $j \neq \hat{j}$ that $\rho_{v'}(i, j) = \rho_v(i, j)$. $\qquad\square$

We now argue $\widetilde{T}$ satisfies item 3 of Definition 5.1. Let $\widetilde{v}$ be a leaf of $\widetilde{T}$ labelled $o \in \mathcal{O}$, and let v be the leaf in T that we reach when we are still at $\widetilde{v}$. By Claim 5.6, we have that $\sigma_v(i) = \rho_v(i, 1) \oplus \ldots \oplus \rho_v(i, m)$ for all $i \in \text{fix}(\rho_v)$. Therefore, since $\rho_{\widetilde{v}}(i) = \sigma_v(i)$, Claim 5.4 implies that every $x \in \{0, 1\}^n$ that extends $\rho_{\widetilde{v}}$ is such that $(x, o) \in \mathcal{S}$, and thus item 3 is satisfied.

To see that $\widetilde{T}$ has width at most $\frac{w}{m-1}$, note that for all nodes v we traverse in T, and for all $i \in [n]$ it holds that if $\sigma_v(i) \neq \star$, then $|\{j \mid \rho_v(i, j) = \star\}| \leq 1$. Indeed, for the root r of T this holds since σ_r is the constant-$\star$ function and for all other nodes v' it holds by the definition of $\sigma_{v'}$. This implies that for all nodes v we traverse in T the partial assignment σ_v has width at most $\frac{w}{m-1}$. The bound on the width of $\widetilde{T}$ follows since for all nodes $\widetilde{v}$ we traverse in $\widetilde{T}$, $\rho_{\widetilde{v}} = \sigma_v$ for some v we traverse in T.

It remains to argue that $\widetilde{T}$ has depth at most $\log s$. To this end, first note that before a variable is queried in $\widetilde{T}$ for the first time, there are $m - 1$ nodes v in T that fall into case 1. Moreover, if $\hat{i}$ is queried in $\widetilde{T}$ when we are at a node v and also at a subsequent node u in T, then there must be at least one node w between v, u such that w falls into case 1. Indeed, first note that there must be some w' in between v and u (including possibly v), such that $|\{j \mid \rho_{w'}(\hat{i}, j) = \star\}| \geq 2$, otherwise $\sigma_u(\hat{i}) \neq \star$ or $|\{j \mid \rho_u(\hat{i}, j) = \star\}| \geq 2$, which contradicts the fact that u falls into case 3. This implies that there must be a w in between w' and u (including possibly w'), that falls into 1 so that $|\{j \mid \rho_u(\hat{i}, j) = \star\}| \leq 1$. This implies that for every query—i.e., every time the depth of $\widetilde{T}$ increases by 1—the size of the subtree of T rooted at the current node v has decreased by at least $1/2$ and thus the depth of $\widetilde{T}$ is at most $\log s$. $\qquad\square$

We conclude this section by noting that, if we increase the threshold $|\{j \mid \rho_{v'}(\hat{i}, j) = \star\}| \leq 1$ to $|\{j : \rho_{v'}(\hat{i}, j) = \star\}| \leq c$ for some $c < m$, a similar proof gives us the following statement.

Theorem 5.7. *Let $\mathcal{S} \subseteq \{0, 1\}^n \times \mathcal{O}$ be a search problem and let $m > c \geq 1$ and $d \geq 1$. If there is a width-w, size-2^{cd} decision tree for $\mathcal{S} \circ \text{XOR}_m^n$, then there is a width-$\left(\frac{w}{m-c}\right)$, depth-$d$ decision tree for $\mathcal{S}$.*

6   Lifting Theorem for Resolution

In this section, we prove a lifting theorem from resolution width to resolution size:

Theorem 2.6 (Restated) (Lifting for resolution). *For any $m, n \geq 1$ and n-variate CNF formula F, if $\mathrm{IND}_m(F)$ has a resolution refutation of size S and depth d, then F has a resolution refutation of width $\lfloor \log_{(m+1)/2} S \rfloor$ and depth d.*

Let F be a CNF formula over variables $z_1, \ldots, z_m$. Recall that we define $\mathrm{IND}_m(F_n)$ to be the CNF formula obtained by substituting in F every occurrence of z_i by

$$(x_{i,1} \to y_{i,1}) \wedge \ldots \wedge (x_{i,m} \to y_{i,m}), \tag{6.1}$$

expanding out to CNF, and including a 3-CNF formula encoding $x_{i,1} \vee \ldots \vee x_{i,m}$ for every $i \in [n]$. We will call the set of variables $\{y_{i,1}, \ldots, y_{i,m}\}$ the *y-block over z_i*, the *ith y-block*, or simply a *y-block*. We say that clause C mentions a y-block if a literal over a variable in that y-block appears in C.

To prove Theorem 2.6, we define a random restriction ρ of the lifted formula (Definition 6.1). We will ensure that a refutation π of the lifted formula becomes a refutation of the original formula $\pi{\restriction}_\rho$, and that the latter is as narrow as claimed in the theorem. Note that restriction certainly does not increase the depth of the refutation.

Definition 6.1 (Random restriction ρ on variables in $\mathrm{IND}(F)$). For every $i \in [n]$, pick a uniform random $j_i \in [m]$. Then set $x_{i,j_i} = 1$ and set $x_{i,j'} = 0$ for all $j' \neq j_i$. Moreover, set the extension variables encoding the clause $x_{i,1} \vee \ldots \vee x_{i,m}$ as a 3-CNF in the way that all clauses in this 3-CNF are satisfied. Finally, set $y_{i,j}$ to $\{0, 1\}$ uniformly at random for each $j \neq j_i$, leaving y_{i,j_i} unassigned.

Note that ρ always fixes all x-variables and extension variables to $\{0, 1\}$, and the restricted formula $\mathrm{IND}_m(F){\restriction}_\rho$ is exactly F after variable renaming $z_i \leftarrow y_{i,j_i}$ for all i. So after applying ρ, a refutation π of $\mathrm{IND}(F)$ becomes a refutation of F. It remains to prove that for some ρ in Definition 6.1, $\pi{\restriction}_\rho$ is of width at most $\lfloor \log_{(m+1)/2} S \rfloor$. We start with a claim bounding the probability that a clauses has width exactly t after restriction.

Claim 6.2. If a clause C over $Vars(\mathrm{IND}(F))$ mentions r many y-blocks, then for any $t \leq r$,

$$\Pr_\rho[w(C{\restriction}_\rho) = t] \leq \binom{r}{t} \left(\frac{1}{m}\right)^t \left(\frac{1 - \frac{1}{m}}{2}\right)^{r-t}. \tag{6.2}$$

Proof. Without loss of generality, assume there are n many z-variables and the y-blocks mentioned in C are over $z_1, \ldots, z_r$. The event $w(C{\restriction}_\rho) = t$ implies that there is an $I \subseteq [r]$, $|I| = t$, such that:

- For every y-block in I, the y-variable unassigned by ρ appears in C, and all other y-literals in C in that block are set to 0. We call such a y-block *contributing*;
- For every y-block in $[r] \backslash I$, all literals in C over y-variables in that block are set to 0. We call such a y-block *non-contributing*.

For each fixed $I \subseteq [r]$ of size t, we show:

$$\Pr_\rho \left[\begin{array}{l} \text{all } y\text{-blocks in } I \text{ are contributing, and} \\ \text{all } y\text{-blocks in } [r] \backslash I \text{ are non-contributing} \end{array} \right] \leq \left(\frac{1}{m}\right)^t \left(\frac{1 - \frac{1}{m}}{2}\right)^{r-t}. \tag{6.3}$$

Then (6.2) follows by a union bound over I.

Recall that ρ consists of independent components $\rho_1, \ldots, \rho_n$ where ρ_i is the part of ρ on variables over z_i. We can write $C = C_1 \vee \ldots \vee C_r \vee C'$ where $C_i = C_{i,x} \vee C_{i,y}$ contains the literals in C over variables over z_i. By independence of the ρ_i's, (6.3) would follow if we show that $\forall i \in [r]$,

$$\Pr_{\rho_i}[y\text{-block } i \text{ is contributing}] \leq \frac{1}{m}, \tag{6.4}$$

$$\Pr_{\rho_i}[y\text{-block } i \text{ is non-contributing}] \leq \frac{1 - \frac{1}{m}}{2}. \tag{6.5}$$

Assume $C_{i,y}$ mentions a many y-variables, then $a \geq 1$. To see (6.4), the probability that one of these y-variables is left intact by ρ_i while others are all set to 0 is $a \cdot \frac{1}{m} \cdot (\frac{1}{2})^{a-1} \leq \frac{1}{m}$. To see (6.5), the probability that all of them are set to 0 is $\frac{m-a}{m} \cdot (\frac{1}{2})^a \leq \frac{1 - 1/m}{2}$. The claim follows as described. $\qquad \square$

Next, we bound the probability that a clause has width at least w after restriction.

Lemma 6.3. *For any clause C over $Vars(\mathrm{IND}(F))$,*

$$\Pr_\rho[w(C{\restriction}_\rho) \geq w] \leq \left(\frac{2}{m+1}\right)^w. \tag{6.6}$$

Proof. Suppose C mentions r many y-blocks ($r \geq w$). By Claim 6.2,

$$\Pr_\rho[w(C{\restriction}_\rho) \geq w] \leq \sum_{i=w}^{r} \binom{r}{i} \left(\frac{1}{m}\right)^i \left(\frac{1-1/m}{2}\right)^{r-i} \tag{6.7}$$

$$= \left(\frac{2}{m+1}\right)^w \sum_{i=w}^{r} \binom{r}{i} \left(\frac{m+1}{m}\right)^i \left(1-\frac{1}{m}\right)^{r-i} \left(\frac{1}{m+1}\right)^{i-w} \left(\frac{1}{2}\right)^{r-i+w} \tag{6.8}$$

$$\leq \left(\frac{2}{m+1}\right)^w \left(\frac{1}{2}\right)^r \sum_{i=w}^{r} \binom{r}{i} \left(1+\frac{1}{m}\right)^i \left(1-\frac{1}{m}\right)^{r-i} \tag{6.9}$$

$$\leq \left(\frac{2}{m+1}\right)^w, \tag{6.10}$$

where we use $\frac{1}{m+1} \leq \frac{1}{2}$ for the second to last inequality, and the Binomial theorem for the last one. $\qquad\square$

With Lemma 6.3 at hand, we can finish the proof of Theorem 2.6.

Proof of Theorem 2.6. Applying a union bound over all clauses in the refutation π of $\mathrm{IND}(F)$ yields

$$\Pr[w(\pi{\restriction}_\rho) > \lfloor \log_{(m+1)/2} S \rfloor] \leq \sum_{C \in \pi} \Pr[w(C{\restriction}_\rho) \geq \lfloor \log_{(m+1)/2} S \rfloor + 1] \tag{6.11}$$

$$\leq S \left(\frac{2}{m+1}\right)^{\lfloor \log_{(m+1)/2} S \rfloor + 1} < 1 \tag{6.12}$$

where we used Lemma 6.3 in the second step. $\qquad\square$

7 Lifting Theorem for Cutting Planes and Circuits via Triangle DAGs

In this section, we prove our triangle-DAG lifting theorem, restated below, by which we obtain our results for cutting planes and monotone circuits as described in Section 2.

Theorem 2.11 (Restated) (Lifting for triangle-DAGs). *Let F be an n-variate unsatisfiable CNF formula, and let $m, w \in \mathbb{N}, \delta > 0$ be arbitrary parameters satisfying $w \leq n$, $0 < \delta < 1 - \frac{1}{\log m}$ and $m \geq (\frac{50n}{\delta})^{2/\delta}$. If there is a triangle-DAG of size $\frac{1}{2} m^{(1-\delta)w}$ and depth d solving $\mathsf{Search}(F) \circ \mathrm{IND}_m^n$, then F has a resolution refutation of width w and depth dw.*

Note that, for large enough m, this lifting theorem is nearly tight. Indeed, if w is the smallest width of any resolution proof of F then there always exists a rectangle-DAG solving $\mathsf{Search}(F) \circ \mathrm{IND}_m^n$ of size at most $2(mn)^w$, which is at most $m^{1.01w}$ if $m \geq n^{300}$. On the other hand, by setting $\delta = \frac{1}{120}$, the lifting theorem implies that there is no triangle-DAG (and hence no rectangle-DAG) of size $m^{0.99w}$.

Our proof of Theorem 2.11 will follow the high-level strategy of [GGKS20], which consists of two steps:

1. For each node in the triangle-DAG, construct a set of clauses of width at most w. The clauses at the root include $\bot$, and the clauses at leaves are weakenings of the initial clauses of F.

2. Argue that the clauses that were constructed for each non-leaf node in the DAG can be derived from the clauses of its children in low width and depth.

The novel aspect of our proof lies in the first step. In previous DAG-like lifting theorems [GGKS20, LMM$^+$22], each triangle is partitioned into a set of *structured rectangles* (from which a low-width clause can be extracted) and an error set. We forgo this approach in favor of covering the triangle by a set of *strips*—a large set of rows of the triangle which can be broken up into *pre-structured* rectangles (from which we can also extract low-width clauses). This allows us to more aggressively identify structured rectangles, decreasing the size of the set of errors. With this improvement, we can complete the proof using two further elements that are largely established in the literature: a Full Image Lemma similar to that in [LMM$^+$22] (with a simpler and self-contained proof), and a careful width analysis in the second step. In addition, our approach naturally gives a unified proof of both the rectangle- and triangle-DAG lifting theorems.

7.1 Set Up: Full Image Lemma and Triangle Lemma

The proof of Theorem 2.11 relies on two technical lemmas: the Full Image Lemma and the Triangle Lemma. Before stating them, we introduce some notation. For sets $A \subseteq B \neq \emptyset$, the *density* of A in B is $|A|/|B|$. For a random variable $\boldsymbol{x}$ over a finite set C, its *min-entropy* is

$$H_\infty(\boldsymbol{x}) := \log \frac{1}{\max_{c \in C} \Pr[\boldsymbol{x} = c]}. \tag{7.1}$$

For a set C, denote by $\boldsymbol{C}$ a random variable distributed uniformly over C, and note that $H_\infty(\boldsymbol{C}) = \log |C|$. For $x \in [m]^n$, $y \in \{0,1\}^{mn} = (\{0,1\}^m)^n$ and $I \subseteq [n]$, we write $x_I \in [m]^{|I|}$ to denote the projection of x, and $y_I \in \{0,1\}^{m|I|}$ to denote the projection of y, to the coordinates I. Similarly, for $A \subseteq [m]^n$ or $A \subseteq \{0,1\}^{mn}$, we denote by A_I the projection of A to I, and $\boldsymbol{A}_I$ denotes the marginal distribution of $\boldsymbol{A}$ on the coordinates I.

For most of this section the structures of $[m]^n$ and $\{0,1\}^{mn}$ are immaterial and we can simply view $[m]^n \times \{0,1\}^{mn}$ as a product of two sets. Under this perspective, we refer to an $x \in [m]^n$ as a *row* and a $y \in \{0,1\}^{mn}$ as a *column*. For any set $S \subseteq [m]^n \times \{0,1\}^{mn}$ we will denote by S^X and S^Y the projection of S to the rows and columns, respectively, that is, $S^X := \{x \in [m]^n \mid (\{x\} \times \{0,1\}^{mn}) \cap S \neq \emptyset\}$, and $S^Y := \{y \in \{0,1\}^{mn} \mid ([m]^n \times \{y\}) \cap S \neq \emptyset\}$. In particular, for any rectangle $R \subseteq [m]^n \times \{0,1\}^{mn}$, we have that $R = R^X \times R^Y$. Given a row $x \in [m]^n$, we denote the set of columns in S along the row x by

$$S[x] := \{y \in \{0,1\}^{mn} \mid \{x\} \times \{y\} \in S\} \tag{7.2}$$

and note that $\{x\} \times S[x] = (\{x\} \times \{0,1\}^{mn}) \cap S$.

For each triangle $T \subseteq [m]^n \times \{0,1\}^{mn}$ we fix some arbitrary choice of a_T and b_T for which $T = \{(x,y) \in [m]^n \times \{0,1\}^{mn} \mid a_T(x) < b_T(y)\}$. For convenience, we arrange all rows in ascending order of a_T from top to bottom and all columns in descending order of b_T from left to right. We call this the *ordering for the triangle T*.

Throughout the proof we consider block-wise partial assignments $\alpha : [n] \to [m] \cup \{*\}$, which we refer to as *pointers*. Let $\mathsf{fix}(\alpha) := \{i \in [n] \mid \alpha^{-1}(i) \neq *\}$. We say $x \in [n]^m$ *is consistent with* α if $x_i = \alpha(i)$ for all $i \in \mathsf{fix}(\alpha)$. For $X \subseteq [n]^m$, we define

$$X_\alpha := \{x \in X \mid x \text{ is consistent with } \alpha\}. \tag{7.3}$$

The proof of the lifting theorem relies on sets that have high min-entropy when restricted to any subset of its coordinates. In the following definition, $\delta > 0$ is a parameter to be specified later.

Definition 7.1. Say that a set $X \subseteq [m]^n$ is α-*dense* if $X \neq \emptyset$ and for every nonempty $I \subseteq [n] \setminus \mathsf{fix}(\alpha)$, $H_\infty((X_\alpha)_I) \geq \delta|I| \log m$. We say $X \subseteq [m]^n$ is α-*predense* if X contains an α-dense subset.

Observe that being α-dense for a fixed α is generally not a monotone property. For instance, the set $[m^\delta]^n$ is α-dense, where α is the constant-$\star$ function, while its union with $\{1\} \times [m]^{n-1}$ is not, provided that $1/\log m < \delta < 1 - 1/n$. By contrast, being predense is by definition a monotone property.

The basic objects in our analysis are rectangles of the following type.

Definition 7.2. A rectangle $R := X \times Y$ is α-*pre-structured* if

    i) X is α-predense; and

    ii) Y satisfies $H_\infty(\mathbf{Y}) \geq mn - m^\delta/(16n)$.

We say that R is α-*structured* if condition i) is replaced with R being α-dense.

    Note that condition ii) is equivalent to saying that the density of Y in $\{0,1\}^{mn}$ is at least $2^{-m^\delta/(16n)}$.

    Similar to previous DAG lifting theorems [GLM$^+$16, GGKS20, LMM$^+$22], ours relies on the following key lemma which states that the image of a structured rectangle is the full Boolean cube. Here, and throughout this section, we omit the subscript m in IND_m, and for $X \subseteq [m]^n$ and $Y \subseteq \{0,1\}^{mn}$, we denote by $\mathrm{IND}(X,Y) \subseteq \{0,1\}^n$ the image of $X \times Y$ under the map IND.

    A self-contained proof of the lemma below is given in Appendix B.

Full Image Lemma. *For any positive integers m and n, any rectangle $X \times Y \subseteq [m]^n \times \{0,1\}^{mn}$, and any parameter $\delta \in (0,1)$ such that $m^\delta \geq (2 \ln 2)n$, it holds that if*

    (1) $H_\infty(\mathbf{X}_I) \geq \delta|I| \log m - 1$ for any nonempty $I \subseteq [n]$, and

    (2) $H_\infty(\mathbf{Y}) \geq mn - m^\delta/(8n)$,

then there exists $x^* \in X$ such that $\mathrm{IND}(\{x^*\}, Y) = \{0,1\}^n$.

    For a partial assignment $\rho \in \{0,1,\star\}^n$, where $I = \mathrm{fix}(\rho)$, define the width-$|I|$ clause $C_\rho := \bigvee_{i \in I} z_i^{1-\rho(i)}$ where $z^1 := z$ and $z^0 := \bar{z}$. In other words, the falsifying assignments of C_ρ in $\{0,1\}^n$, denoted by $C_\rho^{-1}(0)$, comprise precisely the sub-cube specified by ρ. We can interpret the Full Image Lemma as saying that $\mathrm{IND}(\{x^*\} \times Y) = C_\rho^{-1}(0)$, for ρ being the constant-$\star$ assignment. The following corollary is a simple application of the lemma to (almost) pre-structured rectangles.

Corollary 7.3. *Let $m \geq (16n)^{1/\delta}$, let α be a pointer and let $R = R^X \times R^Y$ be a rectangle that satisfies the following two properties (which are a slightly weaker version of Definition 7.2):*

    1. $(R^X)_\alpha$ contains a subset D where $H_\infty(\mathbf{D}_I) \geq \delta|I| \log m - 1$ for all nonempty $I \subseteq [n]\backslash\mathrm{fix}(\alpha)$; and

    2. $H_\infty(R^Y) \geq mn - m^\delta/(16n) - 1$.

If for some $\rho \in \{0,1\}^{\mathrm{fix}(\alpha)}$ it holds that $\mathrm{IND}(x_{\mathrm{fix}(\alpha)}, y_{\mathrm{fix}(\alpha)}) = \rho$ for all $(x,y) \in R$, then there exists an $x^* \in R^X$ such that $\mathrm{IND}(\{x^*\}, Y) = C_\rho^{-1}(0)$.

Proof. If $\mathrm{fix}(\alpha) = [n]$, the conclusion follows from the fact that $R \neq \emptyset$ and for any $(x,y) \in R$, $\mathrm{IND}(x,y) = C_\rho^{-1}(0)$, which is a singleton set. Hence, we assume $|\mathrm{fix}(\alpha)| < n$.

    Let $D \subseteq (R^X)_\alpha$ be a subset witnessing property 1. Let $J := [n] \setminus \mathrm{fix}(\alpha)$. We consider the rectangle $R' := D_J \times R_J^Y \subseteq [m]^{|J|} \times \{0,1\}^{m|J|}$, where D_J and R_J^Y are the projections of D and of R^Y to the coordinates J. We want to show that the Full Image Lemma is applicable to rectangle R'. For this purpose, observe that for any $I \subseteq J$, it follows from the assumption on D that D_J satisfies condition (1) of the lemma. Secondly, note that

$$\frac{|R_J^Y|}{2^{m|J|}} \geq \frac{|R^Y|}{2^{mn}} \geq 2^{-m^\delta/(16n)-1} \geq 2^{-m^\delta/(8|J|)}, \tag{7.4}$$

where the first inequality follows from the definition of projection (i.e., projection does not decrease density), the second inequality follows from item 2, and for the third inequality we use that $m^\delta \geq 16n$. Together, this verifies the conditions in the Full Image Lemma, whose application to R' gives us an element $x^{**} \in D_J$ such that $\mathrm{IND}(\{x^{**}\} \times R_J^Y) = \{0,1\}^{|J|}$. The concatenation of α and x^{**} provides the desired element $x^* \in D \subseteq R^X$. $\square$

In order to extract a resolution proof from a triangle-DAG, we cover a triangle by a set of pre-structured rectangles (from which Corollary 7.3 allows us to extract low-width clauses), along with a small number of "error" rows and columns. Unlike previous approaches which partition the triangle into rectangles, we will cover the triangle with (potentially overlapping) *strips*—sets of pre-structured rectangles which all share the same rows, along with a set of "secured" rows on which can apply Corollary 7.3. This overlapping covering, as opposed to partitioning, allows us to reduce the number of rows and columns which are not within any pre-structured rectangle and hence reduce the error sets.

We now formally define this notion of a strip. For this, let $w \leq n$ be a parameter which corresponds to the width of the resolution proof to be extracted. For a triangle T, recall that $T^X \subseteq [m]^n$ and $T^Y \subseteq \{0,1\}^{mn}$ are the row and column projections of T.

Definition 7.4 (Strips). For a triangle $T \subseteq T^X \times T^Y$ a *strip* S of T is a subset of rows $S \subseteq T^X$ that is α-predense for some pointer $\alpha \in ([m] \cup \{\star\})^n$ with $|\mathsf{fix}(\alpha)| \leq w$. Associated with S are the following:

i) A collection of α-pre-structured rectangles $\mathcal{R}^S = \{R_\beta\}_\beta$ indexed by a set of $\beta \in \{0,1,\star\}^n$ with $\mathsf{fix}(\beta) = \mathsf{fix}(\alpha)$, where each $R_\beta = S \times Y_\beta$ is such that $\mathrm{IND}(\alpha_{\mathsf{fix}(\alpha)}, y_{\mathsf{fix}(\alpha)}) = \beta_{\mathsf{fix}(\alpha)}$ for all $y \in Y_\beta$. Furthermore, within each R_β there is an "inner" sub-rectangle $R_\beta^{\mathsf{in}} \subseteq R_\beta \cap T$ which is α-structured and fully contained within T.

ii) A subset of rows $\widehat{S} \subseteq S$ which we call the rows *secured* by S.

A depiction of a strip is given in Figure 7. The purpose of the secured rows is described by the following lemma, which states that for any triangle T, we can construct a set of strips such that the associated pre-structured rectangles cover all of T except a small set of error rows—rows that are not secured by any strip constructed—and error columns. We note that the definition of strips depends on parameters n, m, w and, due to the definition of α-pre-structured and α-structured, also on δ.

Triangle Lemma. *For any positive integers m, n and $w \leq n$, and parameter $\delta \in (0,1)$ and any triangle $T \subseteq T^X \times T^Y \subseteq [m]^n \times \{0,1\}^{mn}$ there is a set of strips $\mathsf{Strips}(T)$ of T and "error" sets $X_{\mathrm{err}}^T \subseteq [m]^n, Y_{\mathrm{err}}^T \subseteq \{0,1\}^{mn}$ such that for any $x \in T^X$ one of the following cases holds:*

– *Security. If x is secured by a strip $S \in \mathsf{Strips}(T)$, then $\{x\} \times T[x]$ is covered by the rectangles in $\mathcal{R}^S$ together with the error columns, that is,*

$$\{x\} \times T[x] \subseteq \bigcup_{R \in \mathcal{R}^S} R \cup \left(\{x\} \times Y_{\mathrm{err}}^T\right). \tag{7.5}$$

– *Error. If x is not secured by any strip in $\mathsf{Strips}(T)$, then $x \in X_{\mathrm{err}}^T$.*
– *Maximality. If there exists a rectangle $R \subseteq T^X \times (T^Y \setminus Y_{\mathrm{err}}^T)$ that is α-pre-structured for some pointer α with $|\mathsf{fix}(\alpha)| \leq w$ and $\mathrm{IND}(R) \subseteq C_\beta^{-1}(0)$ for some $\beta \in \{0,1,\star\}^n$ with $\mathsf{fix}(\beta) = \mathsf{fix}(\alpha)$, then there exists a strip $S \in \mathsf{Strips}(T)$ with associated pointer α such that $\mathcal{R}^S$ contains a rectangle indexed by β.*

Furthermore, $|X_{\mathrm{err}}^T| \leq m^{n-(1-\delta)w}$ and $|Y_{\mathrm{err}}^T| \leq 2^{mn-m^\delta/(16n)+(w+1)\log(2mn)}$.

We defer the proof of the lemma together with the construction of strips to Section 7.3 in favor of first completing the proof of the lifting theorem.

7.2 Proof of Lifting Theorem

Now we prove Theorem 2.11 using the Triangle Lemma and Corollary 7.3.

Proof of Theorem 2.11. Let Π be any triangle DAG of size $m^{(1-\delta)w}/2$ solving $S \circ \mathrm{IND}_m^n$. We can assume $w \leq n$, otherwise the theorem trivially holds. We first remove the error rows and columns from Π as follows.

Error Removal. Sort the triangles of Π in any topological order $T_1, \dots, T_s$ from the leaves to the root. That is, if T is a child of T' then T comes before T' in the order. We process Π by the following procedure.

Initialize $X_{\text{err}}^0 = Y_{\text{err}}^0 := \emptyset$. For $i = 1, \dots, s$ do the following in order:

1. Remove from T_i the error rows and columns accumulated at $i - 1$, that is,

$$T_i \leftarrow T_i \setminus \left((X_{\text{err}}^{i-1} \times \{0, 1\}^{mn}) \cup ([m]^n \times Y_{\text{err}}^{i-1}) \right). \tag{7.6}$$

2. Let $X_{\text{err}}^{T_i}$ and $Y_{\text{err}}^{T_i}$ be the X- and Y-error sets, respectively, obtained by applying the Triangle Lemma to T_i.

3. Define $X_{\text{err}}^i := X_{\text{err}}^{i-1} \cup X_{\text{err}}^{T_i}$ and $Y_{\text{err}}^i := Y_{\text{err}}^{i-1} \cup Y_{\text{err}}^{T_i}$.

Note that in this procedure, the children nodes will each contribute some error rows/columns to the parents, and every node remains a triangle, as we have only removed whole rows/whole columns from it. Henceforth, Π will refer to the resulting triangle-DAG after this procedure.

We extract from Π a resolution refutation of F by showing that the following two items hold.

– *Clauses.* We can associate with every triangle T in Π a set $\mathcal{C}(T)$ of clauses—each of width at most w—such if T is a leaf of Π then $\mathcal{C}(T)$ is a weakening of an initial clause of F, and if T is the root then the empty clause $\perp$ is contained in $\mathcal{C}(T)$.

– *Inferences.* If triangle T has children T_1 and T_2 in Π then each clause in $\mathcal{C}(T)$ has a width-w and depth-w derivation from the clauses $\mathcal{C}(T_1) \cup \mathcal{C}(T_2)$.

We now prove these items.

Clauses. For each triangle T in Π, apply the Triangle Lemma to obtain a set of strips $\mathsf{Strips}(T)$ of T. We define $\mathcal{C}(T)$ as follows: for each strip $S \in \mathsf{Strips}(T)$ and each pre-structured rectangle $R_\beta \in \mathcal{R}^S$, we include the clause C_β; that is,

$$\mathcal{C}(T) := \bigcup_{S \in \mathsf{Strips}(T)} \left\{ C_\beta \mid R_\beta \in \mathcal{R}^S \right\}. \tag{7.7}$$

To see that C_β is a clause of width at most w, let α with $|\mathsf{fix}(\alpha)| \le w$ be the pointer associated with the strip S. Then Corollary 7.3 guarantees that $\mathrm{IND}(R_\beta) = C_\beta^{-1}(0)$ where the width of C_β is $|\beta^{-1}(0) \cup \beta^{-1}(1)| = |\mathsf{fix}(\alpha)| \le w$.

We now verify that these sets of clauses satisfy the desired root and leaf properties.

– *Root.* Let $R = R^X \times R^Y$ be the triangle at the root of Π (which is a rectangle, though we won't need this). By the Triangle Lemma and a union bound over the triangles in Π, the density X-error accumulated at the root is at most

$$m^{-(1-\delta)w} \cdot |\Pi'| \le m^{-(1-\delta)w} \cdot m^{(1-\delta)w}/2 = 1/2. \tag{7.8}$$

Hence R^X has density at least $1/2$. This implies that for any $\emptyset \ne I \subseteq [n]$,

$$H_\infty \left(R_I^X \right) \ge |I| \log m - 1 \ge \delta |I| \log m, \tag{7.9}$$

and so we have that R^X is $\star^n$-predense.

Similarly, the density of the Y-errors accumulated at the root is at most

$$2^{-m^\delta/(16n) + (w+1)\log(2mn)} \cdot |\Pi| < 2^{-m^\delta/(8n)} < 1/2, \tag{7.10}$$

where in the final inequality we use that $m \ge (50n/\delta)^{\frac{2}{\delta}}$. Therefore, $H_\infty(R^Y \setminus Y_{\text{err}}^R) \ge mn - 1 \ge mn - m^\delta/(16n)$. We therefore conclude that $R^X \times (R^Y \setminus Y_{\text{err}}^R)$ is a $\star^n$-(pre-)structured rectangle.

By the maximality condition of Triangle Lemma applied to $R^X \times (R^Y \setminus Y_{\text{err}}^R)$, we have that there exists a strip $S \in \mathsf{Strips}(R)$ with associated pointer $\star^n$ and such that the collection $\mathcal{R}^S$ is non-empty. By the item i) of Definition 7.4 $\mathcal{R}^S = \{R_\beta\}_\beta$ has to be a singleton set since only the empty string can be a subscript β. Let R_β be the unique rectangle in $\mathcal{R}^S$. Since R_β is $\star^n$-pre-structured, by Corollary 7.3, it holds that $\mathrm{IND}(R_\beta) = \{0, 1\}^n = \perp^{-1}(0)$. Therefore $\perp \in \mathcal{C}(R)$.

– *Leaves.* Consider any leaf triangle T of Π. By definition, there is an axiom clause $C_i \in F$ such that $\text{IND}(T) \subseteq C_i^{-1}(0)$. Therefore, for any clause $C_\beta \in \mathcal{C}(T)$, we have $C_\beta^{-1}(0) = \text{IND}(R_\beta) = \text{IND}(R_\beta^{\text{in}}) \subseteq \text{IND}(T) \subseteq C_i^{-1}(0)$, meaning C_β is a weakening of C_i.

Inferences. Let T be any non-leaf triangle in Π with children T_1 and T_2. Consider any clause $C \in \mathcal{C}(T)$ generated by some pre-structured rectangle $R_\beta = S \times Y_\beta^T$ in a strip S defined from some pointer α. We will show that C has a resolution derivation of width w and depth w from either $\mathcal{C}(T_1)$ or $\mathcal{C}(T_2)$.

Consider the "inner" structured sub-rectangle $R_\beta^{\text{in}} \subseteq R_\beta \cap T$. Since T is covered by its children T_1 and T_2,

$$R_\beta^{\text{in}} \subseteq T \subseteq T_1 \cup T_2. \tag{7.11}$$

We claim that at least one of T_1 or T_2 contains a sub-rectangle $Q = Q^X \times Q^Y \subseteq R_\beta^{\text{in}}$ with X- and Y-density at least half that of R_β^{in}. To see this, order the rows/columns according to the ordering of T_1, then the center p of R_β^{in} divides R_β^{in} into four quadrants. If $p \in T_1$ then, as T_1 is a triangle, the top-left quadrant Q of R_β^{in} is contained entirely within T_1; see Figure 6. Otherwise, if $p \notin T_1$, then as T_1 is a triangle, the bottom-right quadrant Q is disjoint from T_1 and so it must be contained within T_2. In either case, $H_\infty(Q^Y) \geq H_\infty(Y_\beta^T) - 1$ and $H_\infty(Q_J^X) \geq H_\infty(S_J) - 1$ for any $\emptyset \neq J \subseteq [n] \setminus I$. In particular, Q satisfies the premises of Corollary 7.3. Suppose without loss of generality that $Q \subseteq T_1$.
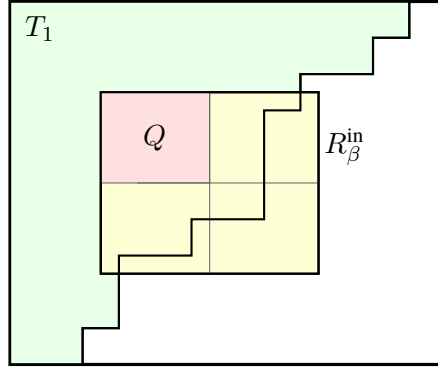

Figure 6: The structured rectangle R_β^{in} for triangle T, whose quadrant Q is contained entirely within child T_1.

Applying Corollary 7.3 to Q, we get a row $x^* \in Q^X \subseteq T_1^X$ such that $\text{IND}(\{x^*\} \times Q^Y) = C_\beta^{-1}(0)$. As we have removed $X_{\text{err}}^{T_1}$ and $Y_{\text{err}}^{T_1}$ from T in the Error Removal step, $x^* \notin X_{\text{err}}^{T_1}$ and $Q^Y \subseteq T^Y$ is disjoint from $Y_{\text{err}}^{T_1}$. Thus, x^* is secured by a strip S' of T_1 defined by some pointer α'. By the Triangle Lemma,

$$\{x^*\} \times Q_Y \subseteq \{x^*\} \times (T_1[x^*] \setminus Y_{\text{err}}^{T_1}) \subseteq \bigcup_{R_\xi \in \mathcal{R}^{S'}} R_\xi, \tag{7.12}$$

where $\mathcal{R}^{S'}$ is the set of pre-structured rectangles in strip S'. By Corollary 7.3, each α'-pre-structured rectangle $R_\xi \in \mathcal{R}^{S'}$ satisfies $\text{IND}(R_\xi) = C_\xi^{-1}(0)$, and so

$$C^{-1}(0) = \text{IND}(R_\beta) = \text{IND}(\{x^*\} \times Q_Y) \subseteq \bigcup_{R_\xi \in \mathcal{R}^{S'}} \text{IND}(R_\xi) = \bigcup_{R_\xi \in \mathcal{R}^{S'}} C_\xi^{-1}(0). \tag{7.13}$$

In particular, this means that C is logically implied by the clauses $\{C_\xi\}$, which depend only on variables in $\text{fix}(\alpha') \subseteq [n]$. Letting C' be the clause obtained from C by discarding the literals over $\text{fix}(\alpha) \setminus \text{fix}(\alpha')$, then C' is also logically implied by the clauses $\{C_\xi\}$. Hence we can derive C in resolution by first deriving C' and then weakening it to C. As the variables of $C' \cup \{C_\xi\}$ are in $\text{fix}(\alpha')$ which has size $\leq w$, and the width of C is at most w, this derivation takes width and depth at most w. Here we have used the fact that any derivation over k variables takes width and depth at most k. $\qquad\square$

7.3 Proof of Triangle Lemma

The rest of this section is dedicated to the proof of the Triangle Lemma. That is, our goal is to describe, for any given parameter w, how to associate with any triangle $T \subseteq T^X \times T^Y$ a set of strips $\mathsf{Strips}(T)$ and error sets which satisfy the *security*, *error* and *maximality* properties of the lemma.

Let parameters w and δ be given. For every pointer α with $|\mathsf{fix}(\alpha)| \leq w$ such that the rows in T^X that are consistent with α form an α-predense set, i.e., $(T^X)_\alpha$ is α-predense, we construct a strip $S := (T^X)_\alpha$, to be included in $\mathsf{Strips}(T)$, by associating S with the following structures.

- *Secured Rows.* Let $x^S \in S$ be the highest row (according to the ordering of T) such that the elements in S above or equal to x^S form an α-predense set. Let the secured rows $\widehat{S} \subseteq S$ be those below or equal to x^S.

- *Pre-Structured Rectangles.* Generate the set of pre-structured rectangles $\mathcal{R}^S$ as follows: for every $\beta \in \{0, 1, \star\}^n$ with $\mathsf{fix}(\beta) = \mathsf{fix}(\alpha)$ consider the set of columns

$$Y_\beta := \{y \in \{0,1\}^{mn} \mid \mathrm{IND}\left(\alpha_{\mathsf{fix}(\alpha)}, y_{\mathsf{fix}(\alpha)}\right) = \beta_{\mathsf{fix}(\alpha)}\}. \tag{7.14}$$

  If $H_\infty(Y_\beta) \geq mn - m^\delta/(16n)$ then we include the rectangle $R_\beta := S \times Y_\beta$ in $\mathcal{R}^S$. Otherwise, we include the columns Y_β in a set Y_{err}^S.

- *Inner Rectangle.* It remains to show that we can find some sub-rectangle $R_\beta^{\mathrm{in}} \subseteq R_\beta \cap T$ which is α-structured and contained entirely within T. Since S is α-predense there is some α-dense subset of rows $S' \subseteq S$. Note that by definition S' is only above (and including) x^S, and so the rectangle $R_\beta^{\mathrm{in}} := S' \times Y_\beta$ is only above (and including) $\{x^S\} \times Y_\beta \subseteq T$. Hence, as T is a triangle, $R_\beta^{\mathrm{in}} \subseteq T$. Finally, note that as R_β was not categorized as "error", R_β^{in} is α-structured.

Observe that with this construction each strip in $\mathsf{Strips}(T)$ is uniquely determined by a pointer α. Finally, define the associated error sets $X_{\mathrm{err}}^T \subseteq [m]^n$ and $Y_{\mathrm{err}}^T \subseteq \{0, 1\}^{mn}$ as follows:

- *X-Error.* Let X_{err}^T be the set of rows in T^X which are not secured by *any* strip in $\mathsf{Strips}(T)$.
- *Y-Error.* Let Y_{err}^T be collected over all strips $S \in \mathsf{Strips}(T)$, that is, $Y_{\mathrm{err}}^T := \bigcup_{S \in \mathsf{Strips}(T)} Y_{\mathrm{err}}^S$.
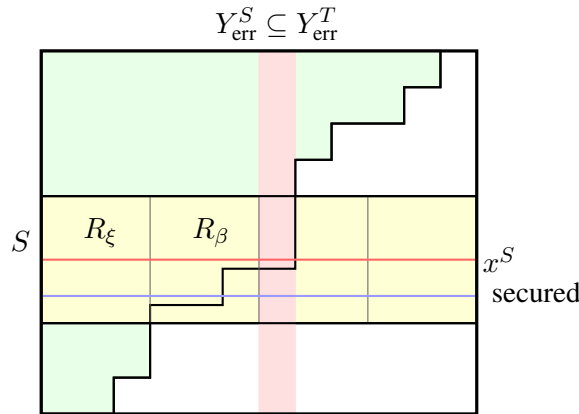
A depiction of a strip is in Figure 7.

Figure 7: A strip S of a triangle, including two pre-structured rectangles R_ξ, R_β, a set of error columns Y_{err}^S, and an example of a secured row.

We first argue that the error, security and maximality properties of the Triangle Lemma hold. The error property holds by construction. To see why the security property holds, note that given a strip $S \in \mathsf{Strips}(T)$, the row x^S in T is covered by the α-pre-structured rectangles and the error columns in S. That is,

$$\{x^S\} \times T[x^S] \subseteq \bigcup_{R \in \mathcal{R}^S} R \cup \left(\{x^S\} \times Y_{\mathrm{err}}^T\right). \tag{7.15}$$

Fix any secured row $x \in \widehat{S} \subseteq S$. Then x is below or equal to x^S and, therefore, since T is a triangle, $T[x] \subseteq T[x^S]$. Hence,

$$\{x^S\} \times T[x] \subseteq \bigcup_{R \in \mathcal{R}^S} R \cup \left(\{x^S\} \times Y_{\mathrm{err}}^T\right) . \tag{7.16}$$

Now, for the maximality property, assume there is a rectangle $R \subseteq T^X \times (T^Y \setminus Y_{\mathrm{err}}^T)$ that is α-pre-structured for some pointer α with $|\mathrm{fix}(\alpha)| \leq w$ and $\mathrm{IND}(R) \subseteq C_\beta^{-1}(0)$ for some $\beta \in \{0, 1, \star\}^n$ with $\mathrm{fix}(\beta) = \mathrm{fix}(\alpha)$. Note that $(R^X)_\alpha$ is α-predense and hence so is its superset $(T^X)_\alpha$, thus by our construction there is a strip $S \in \mathsf{Strips}(T)$ associated with α. Since $\mathrm{IND}(R) \subseteq C_\beta^{-1}(0)$ it follows that R^Y is a subset of $Y_\beta = \{y \in \{0, 1\}^{mn} \mid \mathrm{IND}(\alpha_{\mathrm{fix}(\alpha)}, y_{\mathrm{fix}(\alpha)}) = \beta_{\mathrm{fix}(\alpha)}\}$ defined in (7.14). As $R^Y \cap Y_{\mathrm{err}}^T = \emptyset$ and $R^Y \subseteq Y_\beta$ (and $R^Y \neq \emptyset$ since R is α-pre-structured), it must be the case that $Y_\beta \not\subseteq Y_{\mathrm{err}}^T$ and thus, by the construction of *Pre-Structured Rectangles*, it must be the case that $H_\infty(Y_\beta) \geq mn - m^{\delta/(16n)}$. Therefore, the rectangle $R_\beta := (T^X)_\alpha \times Y_\beta$ is α-pre-structured and thus, by our construction, is in $\mathcal{R}^S$.

Finally, we bound the size of the error sets, using the following two claims.

Claim 7.5. *For any triangle T, the density of X_{err}^T in $[m]^n$ is less than $m^{-(1-\delta)w}$.*

Proof. Suppose for contradiction that X_{err}^T has density at least $m^{-(1-\delta)w}$. For simplicity, we denote $\widehat{X} := X_{\mathrm{err}}^T$. Let $I \subseteq [n]$ be a maximal set of blocks where $\widehat{X}$ is not dense—meaning that $H_\infty(\widehat{\mathbf{X}}_I) < \delta |I| \log m$—and fix any pointer α with $\mathrm{fix}(\alpha) = I$ that witnesses $\Pr[\widehat{\mathbf{X}}_I = \alpha_I] \geq m^{-\delta|I|}$. If no such I exists, we let $I := \emptyset$ and $\alpha = \star^n$. We record the following two basic properties:

(1) $|I| \leq w$,
(2) $\widehat{X}_\alpha := \{x \in X_{\mathrm{err}}^T \mid x_I = \alpha_I\}$ is α-dense.

To see item (1), observe that by the definition of α,

$$|\widehat{X}| \leq \frac{|\widehat{X}_\alpha|}{m^{-\delta|I|}} \leq \frac{|\{x \in [m]^n \mid x_I = \alpha_I\}|}{m^{-\delta|I|}} = m^{n-(1-\delta)|I|} . \tag{7.17}$$

From this and our assumption that $|\widehat{X}|$ has density at least $m^{-(1-\delta)w}$, it follows that $|I| \leq w$. To prove item (2), we show that if $\widehat{X}_\alpha$ is not α-dense then this contradicts the maximality of I. Indeed, if $\widehat{X}_\alpha$ is not α-dense then there exists a nonempty subset $J \subseteq [n] \setminus I$ and a witness $\alpha' \in ([m] \cup \{\star\})^n$ with $\mathrm{fix}(\alpha') = J$ such that $\Pr_{x \sim \widehat{X}_\alpha}[x_J = \alpha'_J] \geq m^{-\delta|J|}$. Let $\alpha \circ \alpha'$ be the pointer with $\mathrm{fix}(\alpha \circ \alpha') = I \cup J$ such that $(\alpha \circ \alpha')_I = \alpha_I$ and $(\alpha \circ \alpha')_J = \alpha_J$. Then

$$\Pr_{x \sim \widehat{X}}[x_{I \cup J} = (\alpha \circ \alpha')_{I \cup J}] = \Pr_{x \sim \widehat{X}}[x_I = \alpha_I] \cdot \Pr_{x \sim \widehat{X}}[x_J = \alpha'_J \mid x_I = \alpha_I] \tag{7.18}$$

$$= \Pr_{x \sim \widehat{X}}[x_I = \alpha_I] \cdot \Pr_{x \sim \widehat{X}_\alpha}[x_J = \alpha'_J] \tag{7.19}$$

$$\geq m^{-\delta(|I|+|J|)}, \tag{7.20}$$

meaning that $\widehat{X}$ is also not dense on $I \cup J$, which contradicts the maximality of I.

By item (1) and item (2) there is a strip $S \in \mathsf{Strips}(T)$ with associated pointer α, consisting of the rows $x \in T^X$ for which $x_I = \alpha_I$. Note that $\widehat{X}_\alpha \subseteq S$, and since $\widehat{X}_\alpha$ is α-predense, the distinguished row x^S of strip S cannot be strictly below all rows in $\widehat{X}_\alpha$. However, this implies that some row $x \in \widehat{X}_\alpha$ is secured by S. This is a contradiction, as $x \in \widehat{X}_\alpha \subseteq X_{\mathrm{err}}^T$ where X_{err}^T contains only rows of T that are not secured by any strip in $\mathsf{Strips}(T)$. $\square$

Claim 7.6. *For any triangle T, the density of Y_{err}^T in $\{0, 1\}^{mn}$ is at most $2^{-m^\delta/(16n)+(w+1)\log(2mn)}$.*

Proof. Each strip S for T is determined by a pointer α with $|\mathrm{fix}(\alpha)| \leq w$, of which there are at most $\sum_{i=0}^w \binom{n}{i} m^i \leq (mn)^{w+1}$ many choices. Moreover, for a fixed α, there are at most $2^{|\mathrm{fix}(\alpha)|} \leq 2^w$ many $\beta \in \{0, 1, \star\}^n$ with $\mathrm{fix}(\beta) = \mathrm{fix}(\alpha)$, and hence at most this many Y_β which are not large, i.e., each contributing at most $2^{mn-m^\delta/(16n)}$ many columns. Hence, $|Y_{\mathrm{err}}^T| < (2mn)^{w+1} 2^{mn-m^\delta/(16n)}$, and so the density of Y_{err}^T can be upper bounded by $2^{-m^\delta/(16n)+(w+1)\log(2mn)}$. $\square$

This completes the proof of the Triangle Lemma.

8   Concluding Remarks

This work opens up many exciting avenues for future research; we end by discussing the ones that we find most intriguing.

Supercritical Trade-offs for Non-monotone Circuits.   We show that supercritical trade-offs exist for monotone circuits. What about for non-monotone circuits? Given that unconditional lower bounds for general circuits are beyond the reach of current techniques, it is interesting to prove the existence of such trade-offs under standard cryptographic assumptions, such as the existence of one-way functions.

Supercritical Trade-offs for Perfect Matching and Tseitin.   Having established truly supercritical trade-offs for monotone circuits and cutting planes, we find it natural to ask for more examples of this phenomenon. As mentioned in the introduction, it is possible that the perfect matching problem exhibits such a trade-off for monotone circuits, and for cutting planes the Tseitin formulas are a candidate. The latter would also resolve the following question.

Separating Stabbing and Cutting Planes.   The quasi-polynomial size cutting planes proof of the Tseitin formulas was obtained by showing that a known upper bound on the Tseitin formulas in a proof system known as *stabbing planes* [BFI$^+$18] could be efficiently translated into cutting planes. In fact, as was shown in [FGI$^+$21], any stabbing planes proof with sufficiently small coefficients can be translated into cutting planes. However, this transformation causes a blow-up in depth that is proportional to the size of the original proof. For example, the depth $O(\log^2 n)$ stabbing planes proofs of the Tseitin formulas become quasi-polynomial-depth cutting planes proofs. Can one show that this blow-up is inevitable by giving a formula which has small stabbing planes proofs with low depth, however exhibits a supercritical size-depth trade-off for cutting planes?

Further Applications of Variable Compression.   We give an application of variable compression in proof complexity. Is it possible to apply this technique to other problems? For example, can *Pebbling formulas* and their associated graphs be compressed? New compressions for the Cop-Robber game would also be of interest.

Acknowledgements

References

[AB87]   Noga Alon and Ravi B. Boppana. The monotone circuit complexity of Boolean functions. *Combinatorica*, 7(1):1–22, March 1987.

[And85]   Alexander E. Andreev. On a method for obtaining lower bounds for the complexity of individual monotone functions. *Soviet Mathematics Doklady*, 31(3):530–534, 1985. English translation of a paper in *Doklady Akademii Nauk SSSR*.

[Bab16]   László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC '16)*, pages 684–697, June 2016.

[BBI16]   Paul Beame, Chris Beck, and Russell Impagliazzo. Time-space tradeoffs in resolution: Superpolynomial lower bounds for superlinear space. *SIAM Journal on Computing*, 45(4):1612–1645, August 2016. Preliminary version in *STOC '12*.

[Ber12]   Christoph Berkholz. On the complexity of finding narrow proofs. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '12)*, pages 351–360, October 2012.

[BFI$^+$18]   Paul Beame, Noah Fleming, Russell Impagliazzo, Antonina Kolokolova, Denis Pankratov, Toniann Pitassi, and Robert Robere. Stabbing planes. In *Proceedings of the 9th Innovations in Theoretical Computer Science Conference (ITCS '18)*, volume 94 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:20, January 2018.

[BGH$^+$06]   Joshua Buresh-Oppenheim, Nicola Galesi, Shlomo Hoory, Avner Magen, and Toniann Pitassi. Rank bounds and integrality gaps for cutting planes procedures. *Theory of Computing*, 2(4):65–90, 2006. Preliminary version in *FOCS '03*.

[BLVS24]   Christoph Berkholz, Moritz Lichter, and Harry Vinall-Smeeth. Supercritical size-width tree-like resolution trade-offs for graph isomorphism. *arXiv preprint arXiv 2407.17947*, July 2024.

[BN20]   Christoph Berkholz and Jakob Nordström. Supercritical space-width trade-offs for resolution. *SIAM Journal on Computing*, 49(1):98–118, February 2020. Preliminary version in *ICALP '16*.

[BN21]   Samuel R. Buss and Jakob Nordström. Proof complexity and SAT solving. In Armin Biere, Marijn J. H. Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, chapter 7, pages 233–350. IOS Press, 2nd edition, February 2021.

[BN23]   Christoph Berkholz and Jakob Nordström. Near-optimal lower bounds on quantifier depth and Weisfeiler-Leman refinement steps. *Journal of the ACM*, 70(5):32:1–32:32, October 2023. Preliminary version in *LICS '16*.

[BNT13]   Chris Beck, Jakob Nordström, and Bangsheng Tang. Some trade-off results for polynomial calculus. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC '13)*, pages 813–822, May 2013.

[BP98]   Paul Beame and Toniann Pitassi. Propositional proof complexity: Past, present, and future. *Bulletin of the European Association for Theoretical Computer Science*, 65:66–89, June 1998.

[BPR95]   María Bonet, Toniann Pitassi, and Ran Raz. Lower bounds for cutting planes proofs with small coefficients. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing (STOC '95)*, pages 575–584, May 1995.

[BT24]   Sam Buss and Neil Thapen. A simple supercritical tradeoff between size and height in resolution. Technical Report TR24-001, Electronic Colloquium on Computational Complexity (ECCC), January 2024.

[CD24]   Arkadev Chattopadhyay and Pavel Dvořák. Super-critical trade-offs in resolution over parities via lifting. Technical Report TR24-132, Electronic Colloquium on Computational Complexity (ECCC), September 2024.

References

[CFI92]   Jin-yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identifications. *Combinatorica*, 12(4):389–410, 1992. Preliminary version in *FOCS '89*.

[Chv73]   Vašek Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4(1):305–337, 1973.

[DT20]   Daniel Dadush and Samarth Tiwari. On the complexity of branching proofs. In *Proceedings of the 35th Annual Computational Complexity Conference (CCC '20)*, volume 169 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 34:1–34:35, July 2020.

[Edm65]   Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[FGI+21]   Noah Fleming, Mika Göös, Russell Impagliazzo, Toniann Pitassi, Robert Robere, Li-Yang Tan, and Avi Wigderson. On the power and limitations of branch and cut. In *Proceedings of the 36th Annual Computational Complexity Conference (CCC '21)*, volume 200 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:30, July 2021.

[FKNP21]   Keith Frankston, Jeff Kahn, Bhargav Narayanan, and Jinyoung Park. Thresholds versus fractional expectation-thresholds. *Annals of Mathematics*, 194(2):475–495, September 2021.

[FP22]   Noah Fleming and Toniann Pitassi. Reflections on proof complexity and counting principles. In Ivo Dünsch and Edwin Mares, editors, *Alasdair Urquhart on Nonclassical and Algebraic Logic and Complexity of Proofs*, volume 22 of *Outstanding Contributions to Logic*, chapter 18, pages 497–520. Springer, 2022.

[FPPR22]   Noah Fleming, Denis Pankratov, Toniann Pitassi, and Robert Robere. Random $\theta(\log n)$-CNFs are hard for cutting planes. *Journal of the ACM*, 69(3):19:1–19:32, June 2022. Preliminary version in *FOCS '17*.

[FPR22]   Noah Fleming, Toniann Pitassi, and Robert Robere. Extremely deep proofs. In *Proceedings of the 13th Innovations in Theoretical Computer Science Conference (ITCS '22)*, pages 70:1–70:23. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, January 2022.

[Für01]   Martin Fürer. Weisfeiler-Lehman refinement requires at least a linear number of iterations. In *Proceedings of the 28th International Colloquium on Automata, Languages and Programming (ICALP '01)*, volume 2076 of *Lecture Notes in Computer Science*, pages 322–333. Springer, July 2001.

[Gál01]   Anna Gál. A characterization of span program size and improved lower bounds for monotone span programs. *Computational Complexity*, 10(4):277–296, December 2001. Preliminary version in *STOC '98*.

[GGKS20]   Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. *Theory of Computing*, 16(13):1–30, 2020. Preliminary version in *STOC '18*.

[GLM+16]   Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. *SIAM Journal on Computing*, 45(5):1835–1869, October 2016. Preliminary version in *STOC '15*.

[GLN23]   Martin Grohe, Moritz Lichter, and Daniel Neuen. The iteration number of the Weisfeiler-Leman algorithm. In *Proceedings of the 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '23)*, pages 1–13, June 2023.

[GLNS23]   Martin Grohe, Moritz Lichter, Daniel Neuen, and Pascal Schweitzer. Compressing CFI graphs and lower bounds for the Weisfeiler-Leman refinements. In *Proceedings of the 64th Annual IEEE Symposium on Foundations of Computer Science (FOCS '23)*, pages 798–809, November 2023.

[GMRS24]   Mika Göös, Gilbert Maystre, Kilian Risse, and Dmitry Sokolov. Supercritical tradeoffs for monotone circuits. Technical report, November 2024.

[GP19]   Erich Grädel and Wied Pakusa. Rank logic is dead, long live rank logic! *The Journal of Symbolic Logic*, 84(1):54–87, March 2019. Preliminary version in *CSL '15*.

[Gro21]   Martin Grohe. The logic of graph neural networks. In *Proceedings of the 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '21)*, pages 1–17, June 2021.

[GTT20]   Nicola Galesi, Navid Talebanfard, and Jacobo Torán. Cops-robber games and the resolution of Tseitin formulas. *ACM Transactions on Computation Theory (TOCT)*, 12(2):1–22, May 2020. Preliminary version in *SAT '18*.

[HP17]   Pavel Hrubeš and Pavel Pudlák. Random formulas, monotone circuits, and interpolation. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS '17)*, pages 121–131, October 2017.

[IL90]   Neil Immerman and Eric Lander. Describing graphs: A first-order approach to graph canonization. In Alan L. Selman, editor, *Complexity Theory Retrospective: In Honor of Juris Hartmanis on the Occasion of His Sixtieth Birthday, July 5, 1988*, pages 59–81. Springer, June 1990.

[Juk12]   Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*, volume 27 of *Algorithms and combinatorics*. Springer, 2012.

[Kie20]   Sandra Kiefer. The Weisfeiler-Leman algorithm: an exploration of its power. *ACM SIGLOG News*, 7(3):5–27, July 2020.

[Kra19]   Jan Krajíček. *Proof Complexity*, volume 170 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, March 2019.

[KS19]   Sandra Kiefer and Pascal Schweitzer. Upper bounds on the quantifier depth for graph differentiation in first order logic. *Logical Methods in Computer Science*, 15(2):19:1–19:19, May 2019. Preliminary version in *LICS '16*.

[KW90]   Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require superlogarithmic depth. *SIAM Journal on Discrete Mathematics*, 3(2):255–265, 1990. Preliminary version in *STOC '88*.

[LMM+22]   Shachar Lovett, Raghu Meka, Ian Mertz, Toniann Pitassi, and Jiapeng Zhang. Lifting with Sunflowers. In *Proceedings of the 13th Innovations in Theoretical Computer Science Conference (ITCS '22)*, volume 215 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 104:1–104:24, January 2022.

[LPS19]   Moritz Lichter, Ilia Ponomarenko, and Pascal Schweitzer. Walk refinement, walk logic, and the iteration number of the Weisfeiler-Leman algorithm. In *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '19)*, pages 1–13, June 2019.

[MLM+23]   Christopher Morris, Yaron Lipman, Haggai Maron, Bastian Rieck, Nils M. Kriege, Martin Grohe, Matthias Fey, and Karsten Borgwardt. Weisfeiler and leman go machine learning: The story so far. *Journal of Machine Learning Research*, 24(333):1–59, 2023.

References

[MRF⁺19] Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI '19)*, pages 4602–4609. AAAI Press, January 2019.

[Pud97] Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *Journal of Symbolic Logic*, 62(3):981–998, September 1997.

[Pud10] Pavel Pudlák. On extracting computations from propositional proofs (a survey). In *Proceedings of the 30th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS '10)*, volume 8 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 30–41, December 2010.

[Rao20] Anup Rao. Coding for sunflowers. *Discrete Analysis*, February 2020.

[Raz85] Alexander A. Razborov. Lower bounds for the monotone complexity of some Boolean functions. *Soviet Mathematics Doklady*, 31(2):354–357, 1985. English translation of a paper in *Doklady Akademii Nauk SSSR*.

[Raz90] Alexander A. Razborov. Applications of matrix methods to the theory of lower bounds in computational complexity. *Combinatorica*, 10(1):81–93, March 1990.

[Raz95] Alexander A. Razborov. Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izvestiya: Mathematics*, 59:205–227, February 1995.

[Raz16] Alexander A. Razborov. A new kind of tradeoffs in propositional proof complexity. *Journal of the ACM*, 63(2):16:1–16:14, April 2016.

[Raz17] Alexander A. Razborov. On the width of semialgebraic proofs and algorithms. *Mathematics of Operations Research*, 42(4):1106–1134, May 2017.

[Raz18] Alexander A. Razborov. On space and depth in resolution. *Computational Complexity*, 27(3):511–559, September 2018.

[RW92] Ran Raz and Avi Wigderson. Monotone circuits for matching require linear depth. *Journal of the ACM*, 39(3):736–744, July 1992.

[Sok17] Dmitry Sokolov. Dag-like communication and its applications. In *Proceedings of the 12th International Computer Science Symposium in Russia (CSR '17)*, volume 10304 of *Lecture Notes in Computer Science*, pages 294–307. Springer, June 2017.

[Sok24] Dmitry Sokolov. Random $(\log n)$-CNF are hard for cutting planes (again). In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24)*, pages 2008—2015, June 2024.

[ST93] Paul D Seymour and Robin Thomas. Graph searching and a min-max theorem for tree-width. *Journal of Combinatorial Theory, Series B*, 58(1):22–33, May 1993.

[Tse68] Grigori Tseitin. On the complexity of derivation in propositional calculus. In A. O. Silenko, editor, *Structures in Constructive Mathematics and Mathematical Logic, Part II*, pages 115–125. Consultants Bureau, New York-London, 1968.

[WL68] Boris Weisfeiler and Andrei Leman. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya, Ser. 2*, 9:12–16, 1968. English translation by Grigory Ryabov available at https://www.iti.zcu.cz/wl2018/pdf/wl_paper_translation.pdf.

A Proof of the Weisfeiler-Leman Result from the Cop-Robber Game

The goal of this appendix is to outline a proof of the trade-off for Weisfeiler-Leman (Theorem 2.2) from our lower bound for the compressed Cop-Robber game (Theorem 4.1). The proof consists of two translations: first from Theorem 4.1 to a round lower bound on a variant of the Cop-Robber game defined below, and then from there to WL-algorithms via the CFI construction [CFI92]. Both steps are standard if there is no compression. With compression, some details should be clarified. We start by recalling some definitions.

A.1 Preliminaries on the Cai–Fürer–Immerman Construction

We recall the CFI-construction [CFI92, Für01]. A graph $G = (V, E)$ is given with the vertices ordered, together with a function $f : E \to \mathbb{F}_2$. We define a colored graph $\mathrm{CFI}(G, f)$ as follows. For each $v \in V$, assume its degree is d, there is a group of 2^{d-1} vertices in $\mathrm{CFI}(G, f)$ identified by $(v, \vec{a})$ for each $\vec{a} = (a_1, \ldots, a_d) \in \mathbb{F}_2$ such that $a_1 + \cdots + a_d = 0$ in $\mathbb{F}_2$, and these 2^{d-1} vertices are all colored v. The vertex order induces an adjacency list form of G. For any edge $e = \{u, v\}$ in G, assuming that u is the ith neighbor of v and v is the jth neighbor of u, we add an edge between $(u, \vec{a})$ and $(v, \vec{b})$ if $a_i + b_j = f(e)$. Finally, the color of vertex $(v, \vec{a})$ is v.

The color-preserving automorphisms of $\mathrm{CFI}(G, f)$ are characterized by *twistings* [CFI92, GP19]. For a set of edges F we denote by $\vec{F}$ the set of directed edges $\{(u, v), (v, u) \mid \{u, v\} \in F\}$. We call $T \subseteq \vec{E}(G)$ a *G-twisting* (or *twisting* in short) if $|T \cap (\{v\} \times V)|$ is even for every $v \in V(G)$. An edge $\{u, v\} \in E(G)$ is *twisted by T*, if $(u, v) \in T$ and $(v, u) \notin T$, or vice-versa. A vertex $v \in V(G)$ is *fixed by T* if $T \cap (\{v\} \times V) = \emptyset$.

Next, we construct the compressed CFI graphs [GLNS23]. Assume G is a vertex ordered graph and $\equiv_V$ is a compatible vertex equivalence relation on $V(G)$ (recall this means $v \equiv_V v'$ implies that v, v' are non-adjacent and have the same degree). Assume in addition that $\equiv_V$ satisfies the following condition: if $u \equiv_V u'$ and $v \equiv_V v'$, and v is the ith neighbor of u, then v' is the i the neighbor of u'.[3] We call a function $f : E(G) \to \mathbb{F}_2$ *compressible* if for all $u \equiv_V u'$ and $v \equiv_V v'$ where both $\{u, v\}$ and $\{u', v'\}$ are edges, $f(\{u, v\}) = f(\{u', v'\})$. Given such a G, such a vertex equivalence $\equiv_V$, and a compressible f, the *compressed CFI graph* $\mathrm{CFI}(G, f)/\equiv_V$ is the *quotient* graph of $\mathrm{CFI}(G, f)$ by the vertex equivalence relation under which $(u, \vec{a})$ is equivalent to $(v, \vec{b})$ if and only if $u \equiv_V v$ and $\vec{a} = \vec{b}$. We denote this vertex equivalence on $\mathrm{CFI}(G, f)$ still by $\equiv_V$. The color of vertices in $\mathrm{CFI}(G, f)/\equiv_V$ can be defined in various canonical ways, e.g., coloring $(u, \vec{a})/\equiv_V$ by the minimum $v \in V(G)$ in class $u/\equiv_V$.

The color-preserving automorphisms of $\mathrm{CFI}(G, f)/\equiv_V$ are characterized by *compressed twistings*, which are G-twistings T where, for any (u_1, v_1) and (u_2, v_2) such that $u_1 \equiv_V u_2$ and v_1 has the same order in the neighbor-list of u_1 as v_2 does in that of u_2, it holds that $(u, v) \in T \Leftrightarrow (u', v') \in T$.

For the cylinder graph G_{cyl}, we use the lexicographical vertex order on $[k] \times [2r + L]$. The associated adjacency-list form of G_{cyl} induces an edge equivalence from $\equiv_V$ in Definition 3.1, which is the same as $\equiv_E$ in Definition 3.4.

Finally, the argument uses a game variant where the Robber is on the edges. Given a graph compression $(G, \equiv_V, \equiv_E)$ as in Definition 3.1, the *compressed k-Cop-edge-Robber* game proceeds as in Definition 3.11, except that the Robber now stays on an edge, and the rule (G2) becomes that the Robber should provide a compressible twisting T which twists only the edge he currently occupies and the edge he moves to, and T fixes every vertex that is $\equiv_V$-equivalent to a Cop position. The game ends if both endpoints of the Robber's edge are $\equiv_V$-equivalent to some Cop positions.

A.2 Proof of Translation from the Cop-Robber Game to the Weisfeiler–Leman Algorithm

We restrict our attention to the compression $(G_{\mathrm{cyl}}, \equiv_V, \equiv_E)$ in Theorem 4.1. For ease of notation, we will call the game in Definition 3.11 the *compressed k-VR* (indicating vertex Robber), and the above game variant the *compressed k-ER* (indicating edge Robber). We say a vertex $v \in V(G)$ is *singleton-class* if

---

[3]For G_{cyl} with the lexicographical vertex order on $[k] \times [2r + L]$ and the vertex equivalence relation $\equiv_V$ in Definition 3.4, this condition is satisfied.

it forms a singleton $\equiv_V$-class, i.e., $|v_{\equiv_V}| = 1$. Similarly, we say an edge $e \in E(G)$ is *singleton-class* if $|e_{\equiv_E}| = 1$.

Lemma A.1 says that our Robber strategy in Section 4.3 can be carried over to the compressed game with edge Robber.

Lemma A.1. *Assume that the Robber has a winning strategy in the R-round k-VR on $(G_{\mathrm{cyl}}, \equiv_V, \equiv_E)$, where the Robber always occupies a vertex $v \in V$ that is singleton-class and such that there are two edges $e, e' \in E$ incident to v that are singleton-class. Then there is a winning strategy for the Robber in the R-round k-ER on G and $\equiv_V$, where the Robber always occupies as singleton-class edge incident to a singleton-class vertex.*

Proof. We fix a strategy S for Robber that wins the R-round k-VR, and by simulating it, we will construct a winning strategy for the Robber in the R-round k-ER. During the simulation, we maintain the invariant that the Cops occupy the same vertices in both graphs, that the Robber in k-VR is on a singleton-class vertex w_1 with two incident singleton-class edges e_1, e_1', and that the Robber in k-ER is on either e_1 or e_1'. In the initial round of the game the invariants hold by assumption, where we can assume without loss of generality that the Robber is placed on such an edge in the beginning.

Assume that the invariants hold, that the game has lasted for $R' < R$ rounds, and that the Cops are playing step 1 (one Cops is picked up, and a destination $x \in V(G)$ selected). In the k-VR, the same Cops is lifted up, and x is signaled to the Robber. As $R' < R$, the Robber in the k-VR has a compressible move $M \subseteq E(G)$ according to the strategy S from w_1 to w_2, where w_2 is singleton-class and has two incident singleton-class edges e_2, e_2'. Without loss of generality, we may assume that the Robber is on e_1 in the k-ER. We want to provide a $\equiv_V$-compressible G-twisting T that only twists edges e_1 and e_2 (or possibly e_2' instead of e_2), and fixes every vertex in a $\equiv_V$-class occupied by a Cops.

First, we assume that w_1 and w_2 are not adjacent. For all $v \in V(M) \setminus \{w_1, w_2\}$, we include in T all edges in M incident to v in the outgoing direction from v, i.e., for any edge $\{v, v'\} \in M$, we include (v, v') in T. For w_1 and w_2, we do the same, except for the edges e_1 and e_2. If e_1 (or e_2) is in M, we do not include it in T in the outgoing direction from w_1 or w_2, respectively. Otherwise (that is, $e_1 \notin M$ or $e_2 \notin M$), we include e_1 or e_2 in the outgoing direction from w_1 or w_2 in T, respectively. T is a twisting, as all $v \in V$ satisfy that $T_v = T \cap (\{v\} \times V)$ is of even size. (For $v \notin V(M)$, $T_v = \emptyset$, for $v \in V(M) \setminus \{w_1, w_2\}$, $|T_v| = \deg_M(v)$, which is even by (G2)(c), and for w_1, w_2, $|T_v| = \deg_M(v) \pm 1$, which is even by (G2)(c).) T is $\equiv_V$-compressible, as M is closed under $\equiv_E$ by (G2)(a), and the directed edges in exactly one of T and $\vec{M}$ are singleton-class. Now T twists exactly e_1 and e_2 by construction, and all the vertices in a $\equiv_V$-class by a Cops are fixed due to (G2)(b).

Let us now turn to the case where w_1 and w_2 are adjacent. Recall that the Robber is on e_1. If $e_1 \neq \{w_1, w_2\}$, let $e' \in \{e_2, e_2'\}$ be such that $e' \neq \{w_1, w_2\}$, then we can use the same twisting T as above, twisting only e_1 and e'. So assume that $e_1 = \{w_1, w_2\}$. Without loss of generality, we assume that $e_2 \neq e_1$ (otherwise switch the roles of e_2 and e_2'). To construct T, we treat $v \in V(M) \setminus \{w_2\}$ as above, and immediately get that T_v is of even size for $v \neq w_2$. For w_2, we include in T all incident edges in M except e_1 and e_2, in the outgoing direction from w_2. If $e_1 \in M$, we include it in T in the outgoing direction from w_2. If $e_2 \notin M$, we include it in T in the outgoing direction from w_2. To see that $|T_{w_2}|$ is even, note that if $e_2 \in M$, $|T_{w_2}| = \deg_M(w_2) - 1$, and otherwise $|T_{w_2}| = \deg_M(w_2) + 1$. T is $\equiv_V$-compressible, twists only e_1 and e_2, and fixes the vertices in the $\equiv_V$-class of the Cops for the same reasons as above.

In the k-ER, the Robber moves according to the twisting T constructed above. Finally, a Cops is placed on x in step 3 of the k-ER. The same happens in the k-VR. As the Robber is not caught in the k-VR, the Robber is also not caught in the k-ER, Since the invariants are maintained throughout the simulation, this concludes the proof of this lemma. □

Proof of Theorem 2.2. For clarity, we will use N for the number of vertices in each graph in the pair, leaving n as the parameter in the graph compression in Definition 3.9. We follow the standard chain of reasoning as in [GLNS23], pointing out necessary changes. Given k, c as in the theorem statement, we take n to be large enough such that the conditions of Definition 3.9 are satisfied. We apply the CFI construction with $f, g \colon E(G_{\mathrm{cyl}}) \to \mathbb{F}_2$, where f is the all-zero function and g is 1 only on one edge adjacent to vertex

$(1, 1)$. This gives us graphs $G_N := \mathrm{CFI}(G_{\mathrm{cyl}}, f)/\equiv_V$ and $H_N := \mathrm{CFI}(G_{\mathrm{cyl}}, g)/\equiv_V$, where N denotes the vertex set size of both. We observe that $N \leq 2^{4-1}\left(2r + k^2(2n)^{c+1}\right) < 2^{c+5}k^2n^{c+1}$.

Using Lemma A.1 and Theorem 4.1, we get their Lemma 29 and consequently Theorem 30 with $k + 1$ replaced by $k + c$, which says that $(k + c - 1)$-WL requires at least $(L - 2r)/(8(k + c)) \geq n^k/(32k)$ rounds to distinguish G_N and H_N. This is at least $\left(2^{-(c+10)}k^{-3}N\right)^{k/(c+1)}$. □

B Proof of Full Image Lemma

In this appendix, we give a self-contained proof of the Full Image Lemma, restated below.

Full Image Lemma. *For any positive integers m, n, rectangle $X \times Y \subseteq \{m\}^n \times \{0, 1\}^{mn}$, and parameter $\delta \in (0, 1)$, assume $m^\delta \geq \frac{4}{\ln 2}n$,*

(1) $H_\infty(\boldsymbol{X}_I) \geq \delta |I| \log m - 1$ for any nonempty $I \subseteq [n]$,

(2) $H_\infty(Y) \geq mn - m^\delta/(8n)$.

There exists $x^* \in X$ such that $\mathrm{IND}(\{x^*\}, Y) = \{0, 1\}^n$.

We claim no originality here, as the argument is the same as in [LMM$^+$22]. Except that we substitute their use of strong sunflower lemmas with a simpler result from [FKNP21] proved by Janson's inequality.

Given a set U, a set sequence $(S_1, \ldots, S_l)$ is κ-*spread over* U if each S_i is a subset of U and for any $W \subseteq U$, the number of elements in the sequence that contains W is at most $l\kappa^{-|W|}$.

Proposition B.1 (Lemma 3.2 of [FKNP21]). *Suppose $(S_1, \ldots, S_l)$ is a sequence of size-r sets that is κ-spread over U. For any $p \in (0, 1)$, if W is a random subset of U where each element is included independently with probability p, then*

$$\Pr_W \left[(\forall i \in [l])\, S_i \not\subseteq W\right] \leq \exp\left(-\frac{p\kappa}{r}\exp(-\frac{r-1}{p\kappa})\right). \tag{B.1}$$

In particular, if $p\kappa \geq (r - 1)/\ln 2$, then the bound in (B.1) can be replaced by $\exp(-p\kappa/(2r))$.

Proof of Proposition B.1. Denote by ξ_i the indicator variable of the event $S_i \subseteq W$. Let

$$\mu := \sum_{i=1}^{l} \mathbb{E}[\xi_i] \tag{B.2}$$

$$\Lambda := \sum_{(i,j):\, S_i \cap S_j \neq \emptyset} \mathbb{E}[\xi_i \cdot \xi_j] = \sum_{i \in [l]}\, \sum_{j:\, S_j \cap S_i \neq \emptyset} \mathbb{E}[\xi_i \cdot \xi_j] \tag{B.3}$$

Then $\mu = l \cdot p^r$, and we can upper bound Λ as follows. For each inner sum in (B.3), we group the terms according to $a := |S_j \cap S_i|$, where in each group there are $\binom{r}{a}$ many choices of $S_j \cap S_i$, and for each choice there are at most $l\kappa^{-a}$ many possible j due to spreadness. So

$$\Lambda \leq l \cdot \sum_{a=1}^{r} \binom{r}{a}(l\kappa^{-a})p^{2r-a} = \mu^2\left((1 + \frac{1}{p\kappa})^r - 1\right) \leq \mu^2 \cdot \frac{r}{p\kappa}\exp(\frac{r-1}{p\kappa}) \tag{B.4}$$

where the last step uses $(1 + x)^r - 1 \leq x \cdot re^{(r-1)x}$. By Janson's inequality applied to $(\xi_1, \ldots, \xi_l)$,

$$\Pr_W \left[(\forall i \in [l])\, \xi_i = 0\right] \leq \exp(-\frac{\mu^2}{\Lambda}) \leq \exp\left(-\frac{p\kappa}{r}\exp(-\frac{r-1}{p\kappa})\right). \qquad \square$$

Remark B.2. The probability bound in (B.1) was improved by [Rao20, Lemma 4] to $r \cdot \exp(-p\kappa/C_{\mathrm{abs}})$ if $p < \frac{1}{2}$, where C_{abs} is an absolute constant. (There, a slightly different notion of r-spreadness and distribution of W is used, but the argument is adaptable.) We use (B.1) for simplicity, which is sufficient for our purpose (we don't intend to optimize gadget size).

Proof of Full Image Lemma. Given $X \subseteq [m]^n$ as in the lemma, we consider the maximum possible size of $Y' \subseteq \{0,1\}^{mn}$ subject to the following condition:

$$(\forall x \in X)\, (\exists z_x \in \{0,1\}^n)\, z_x \notin \mathrm{IND}(\{x\} \times Y'). \tag{B.5}$$

As the first step, we show that for the purpose of maximizing $|Y'|$, we can assume $z_x = \vec{1}$ for all $x \in X$ in (B.5). Then, as the second step, we use Proposition B.1 to show $|Y'| < 2^{mn} 2^{-m^\delta/(8n)}$ in that case. Given our assumption (2) on $|Y|$, this means (B.5) cannot hold for $Y' \leftarrow Y$, so the lemma follows.

For the first step, take mn boolean variables $p_{i,j}$ ($i \in [n]$, $j \in [m]$). For each $x \in [m]^n$, we take a clause $C_x := \bigvee_{i=1}^{n} p_{i,x(i)}^{1-z_x(i)}$, where $p^1 := p$ and $p^0 := \bar{p}$. Then condition (B.5) equivalently says that each element in Y is a satisfying assignment of the CNF $C := \bigwedge_{x \in X} C_x$. We have the following:

Fact B.3 ([LMM$^+$22], Claim 8). *For any CNF F, let F^{mon} be its monotonization by negating each positive literal. Then $|\mathrm{sat}(F)| \leq |\mathrm{sat}(F^{mon})|$, where $\mathrm{sat}(\cdot)$ means the set of satisfying assignments.*

To see this, note that F^{mon} can be obtained from a sequence $F_0 := F, F_1, \ldots, F_N = F^{mon}$ where F_i monotonize F_{i-1} at the ith variable, and we only need to show that $|\mathrm{sat}(F_{i-1})| \leq |\mathrm{sat}(F_i)|$. For this aim, we fix an $i \in [n]$ and view $\{0,1\}^n$ as a collection of pairs $P_y = \{(y,0), (y,1)\}$ over $y \in \{0,1\}^{[n] \setminus \{i\}}$. For each y, if $P_y \subseteq \mathrm{sat}(F_{i-1})$ then the partial assignment y already satisfies each clause in F_{i-1}, so $P_y \subseteq \mathrm{sat}(F_i)$; if exactly one of $(y,0)$ and $(y,1)$ satisfies F_{i-1}, then $(y,0)$ satisfies F_i. Thus for each y, $|\mathrm{sat}(F_{i-1}) \cap P_y| \leq |\mathrm{sat}(F_i) \cap P_y|$. Consequently, $|\mathrm{sat}(F_{i-1})| \leq |\mathrm{sat}(F_i)|$, and Fact B.3 follows.

Fact B.3 says that for the purpose of maximizing $|Y|$, we can take $z_x = \vec{1}$ for all x in condition (B.5). So below we fix $z_x = \vec{1}$ and upper bound $|Y|$. The argument uses a translation of language as follows. Take a ground set $U := [mn]$, understood as the union of n disjoint sets each having size m; we call these m sets m 'groups' for clarity. We will consider set systems over U. Each $x \in [m]^n$ corresponds to a subset $\underline{x} \subseteq U$ which contains one element per group. Each $y \in Y$, when read as a function from U to $\{0,1\}$, corresponds to the subset $\underline{y} := y^{-1}(1) \subseteq U$. Then, assuming $z_x = \vec{1}$ for all x, condition (B.5) becomes

$$(\forall y \in Y)(\forall x \in X)\, \underline{x} \not\subseteq \underline{y}. \tag{B.6}$$

The following fact provides the last bit of the translation, proved by a direct inspection of definitions.

Fact B.4. *For any $A \subseteq [m]^n$ and $\kappa \geq 1$, $\underline{A} = \{\underline{a} \mid a \in A\}$ is κ-spread if and only if $H_\infty(\mathbf{A}_I) \geq |I| \log \kappa$ for all nonempty $I \subseteq [n]$.*

By Fact B.4 and the assumption (1) on X, the set family $\underline{X}$ is $\frac{1}{2}m^\delta$-spread. So we can apply Proposition B.1 with parameters $r := n$, $\kappa := \frac{1}{2}m^\delta$, $p := \frac{1}{2}$, where it holds that $p\kappa = m^\delta/4 \geq r/\ln 2$, so the "in particular" part of the proposition applies. As the result, (B.6) implies that $|Y| < 2^{mn} 2^{-m^\delta/(8n)}$. Then by our first step, (B.5) also implies $|Y| < 2^{mn} 2^{-m^\delta/(8n)}$. But $|Y| \geq 2^{mn} 2^{-m^\delta/(8n)}$ by the assumption (2), so (B.5) cannot hold, i.e., $\exists x^* \in X$ such that $\mathrm{IND}(\{x^*\} \times Y) = \{0,1\}^n$. The Full Image Lemma follows. $\qquad\square$