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

Susanna F. de Rezende
Lund University
Lund, Sweden
susanna.rezende@cs.lth.se

Noah Fleming
Memorial University of Newfoundland
St. John's, Canada
nfleming@mun.ca

Duri Andrea Janett
University of Copenhagen
Copenhagen, Denmark
Lund University
Lund, Sweden
duja@di.ku.dk

Jakob Nordström
University of Copenhagen
Copenhagen, Denmark
Lund University
Lund, Sweden
jn@di.ku.dk

Shuo Pang
University of Copenhagen
Copenhagen, Denmark
shpa@di.ku.dk

Abstract

We exhibit supercritical trade-off for monotone circuits, showing that there are functions computable by small circuits for which any small circuit must have depth superlinear 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 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.

CCS Concepts

• **Theory of computation → Proof complexity**; *Circuit complexity*; *Finite Model Theory*.

Keywords

proof complexity, monotone circuit complexity, supercritical trade-off, resolution, cutting planes, Weisfeiler–Leman algorithm

ACM Reference Format:
Susanna F. de Rezende, Noah Fleming, Duri Andrea Janett, Jakob Nordström, and Shuo Pang. 2025. Truly Supercritical Trade-Offs for Resolution, Cutting Planes, Monotone Circuits, and Weisfeiler–Leman. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC '25), June 23–27, 2025, Prague, Czechia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3717823.3718271

This work is licensed under a Creative Commons Attribution 4.0 International License.
STOC '25, Prague, Czechia
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1510-5/25/06
https://doi.org/10.1145/3717823.3718271

1 Introduction

Computational complexity aims to understand the amount of different resources—such as running time or memory—required in order to solve computational problems. An important question is to understand how 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 [52] showed 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 [4, 8, 9, 11, 15, 18, 26, 52–54] (including the works [4, 8, 9] predating [52]) and finite model theory [12, 33]. Recent papers [23, 26, 30] have raised the question of whether there are supercritical trade-offs in circuit complexity. In this work, we give an affirmative answer to this question.

1.1 Trade-offs in Circuit Complexity

In circuit complexity, different measures of the cost associated with computing a Boolean functions with families of Boolean circuits, such as *size* or *depth*, are of interest. The size of a circuit is the number of gates in it, and depth refers to the longest path from input to output. An intriguing challenge in complexity theory is presented by the perfect matching problem. Although it has been known to be solvable in polynomial time for nearly 70 years [22], many questions about this problem remain unresolved, in particular,

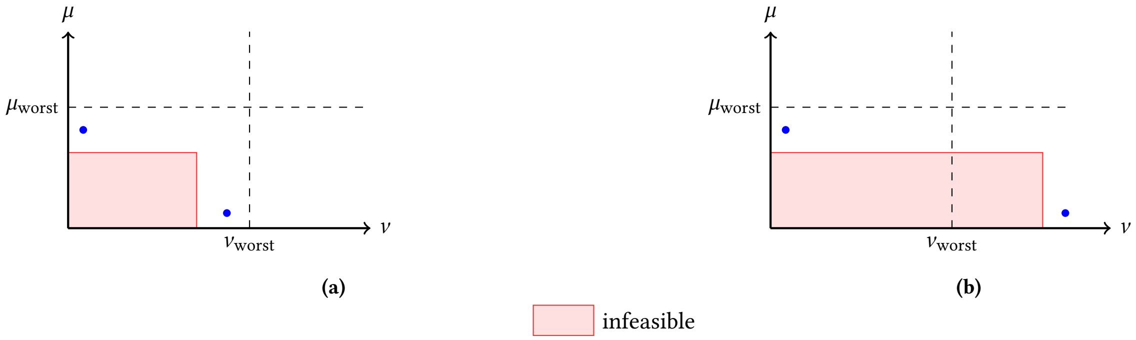


Figure 1: An illustration of trade-offs. Blue dots represent provable upper bounds on measures μ and ν. Proofs with measures in the shaded region are ruled out by the trade-off, where μ_{worst} and ν_{worst} are the worst-case upper bound on μ and ν, respectively. Figure 1a illustrates a non-supercritical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

regarding its monotone complexity. In a breakthrough result in 1985, Razborov [49] proved the first superpolynomial size lower bound for monotone circuits—Boolean circuits with only AND and OR gates, and no negations—for two functions: k-clique and bipartite perfect matching. (Independently, Andreev [2] showed an exponential size lower bound for a different function.)

A few years later, Alon and Boppana [1] improved the lower bound for k-clique to exponential for large k. For bipartite perfect matching, however, Razborov's lower bound remains the state of the art. Raz and Wigderson [48] 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 $O(n)$ and size $2^{O(n)}$. In fact, this rather straightforward upper bound remains the best known to this day. Are there monotone circuits computing bipartite perfect matching in size $n^{O(\log n)}$? 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 already know 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 superlinear depth? It is natural to ask, as was done in [23, 26, 30], if there are any monotone functions that exhibit this kind of supercritical trade-off behavior, where small circuits exist but any small circuit requires superlinear 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 (informal)). *There are N-variate Boolean functions f_N such that:*

(1) f_N is computable by monotone Boolean circuits with size s polynomial in N, but any monotone real circuit with size at most $s^{1.4}$ computing f_N must have depth at least $N^{2.4}$.

(2) f_N is computable by monotone Boolean circuits with size s quasi-polynomial in N, but any monotone real circuit with size at most $s \cdot \exp\left((\log N)^{1.9}\right)$ computing f_N must have depth superpolynomial in N.

The functions we present that exhibit this behavior are obtained from new supercritical trade-offs in the neighboring field of proof complexity, which we discuss next.

1.2 Trade-offs in Proof Complexity

Proof complexity studies how efficient certificates of unsatisfiability of formulas in *conjunctive normal form* (CNF) can be. In this study, the *Tseitin formulas*—unsatisfiable systems of mod 2 linear equations—have played a pivotal role. In particular, they were used to prove the first proof complexity lower bounds in [58] (for a restricted version of the well-studied *resolution* proof system). Since then, the Tseitin formulas have been central in understanding the reasoning power of proof systems, and, in particular, in establishing lower bounds for them; see [25] for a survey. A particularly intriguing case in this regard is the *cutting planes* proof system, which captures reasoning in terms of $\{0, 1\}$-linear inequalities. The first paper studying the complexity of cutting planes proofs [19] conjectured that the Tseitin formulas were hard to prove in cutting planes, and this was reiterated in [7, 37]. While lower bounds on the size of cutting planes proofs have been established for a variety of formulas [13, 24, 30, 35, 46, 57], determining the complexity of cutting planes refutations of Tseitin formulas remained open.

In a surprising turn of events, Dadush and Tiwari [20] exhibited small (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 [5, 23, 26], which would give a partial explanation as to why these proofs took so long to find.

Progress on this question was made in [14, 23], by showing that any cutting planes proof of the Tseitin formulas on n variables requires depth $\Omega(n)$; and in [15, 26], 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 [20], which is supercritical in terms of the formula size as well. In this work, we will refer to trade-offs that are supercritical in the input size—rather than in the number of variables—as *truly* supercritical. We give the first truly supercritical size-depth trade-offs for cutting planes.

Theorem 1.2 (Cutting planes trade-offs (informal)). *There are* 3*-CNF formulas F_N of size $S(F_N)$ over N variables such that:*

1. *Resolution refutes F_N in size $S(F_N \vdash\perp)$ polynomial in N, but any cutting planes refutation with size at most $S(F_N \vdash\perp)^{1.4}$ must have depth at least $S(F_N)^{2.4}$.*
2. *Resolution refutes F_N in size $S(F_N \vdash\perp)$ quasi-polynomial in N, but any cutting planes refutation with size at most $S(F_N \vdash\perp)\cdot \exp\big((\log N)^{1.9}\big)$ must have depth at least superpolynomial in N.*

As cutting planes is stronger than resolution, the theorem also implies the first truly supercritical size-depth trade-offs for resolution. Except for [4, 8, 9], no such results were known. In this work, we also obtain truly supercritical trade-offs for other combinations of complexity measures, which we state next.

First, we restrict our attention to resolution proofs that look like trees. The seminal work [52] 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 (informal)). *There are CNF formulas F_N of size $S(F_N) = \mathrm{poly}(N)$ over N variables such that:*

1. *Resolution refutes F_N in width $W(F_N \vdash\perp) = o(\log N)$, but any treelike refutation with width at most $1.4 \cdot W(F_N \vdash\perp)$ must have size at least $\exp\big(S(F_N)^{2.4}\big)$.*
2. *Resolution refutes F_N in width $W(F_N \vdash\perp) = o\big((\log N)^{3/2}\big)$, but any treelike resolution refutation with width at most $W(F_n \vdash\perp)+40\frac{\log N}{\log\log N}$ must have size at least $\exp\big(S(F_N)^{\omega(1)}\big)$.*

Underlying each of these results is the first truly supercritical width-depth trade-off that is non-trivially robust—we use this to obtain our other trade-offs by applying several (new or improved) lifting theorems.

Theorem 1.4 (Width-depth trade-offs (informal)). *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 with width $w = \lfloor \frac{n}{2\ln n} \rfloor + 3$, such that:*

1. *$N = \mathrm{poly}(n)$ and any refutation of width at most $w + C$ has depth exponential in $\mathrm{poly}(S(F_N))$.*
2. *$N = o(2^{n/2})$ and any refutation of width at most $(1+\delta)w$ has depth superlinear in $S(F_N)$.*

Prior to our work, the only truly supercritical trade-off for width versus depth was due to Berkholz [9]. However, the trade-off there has no robustness (i.e., it holds only for the minimum width) and so cannot be used to obtain other trade-offs.

1.3 Trade-offs for Weisfeiler–Leman

Perhaps somewhat 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 [3], is related to machine learning [32, 44, 45], and has also turned out to be relevant for other areas [34, 39]. 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 [33, 40, 42]. For a long time, the best lower bound was linear [27] until the works [12, 33] 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 [34] 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. We resolve this problem, and our robust trade-off for Weisfeiler–Leman is the foundation for the other results in this paper.

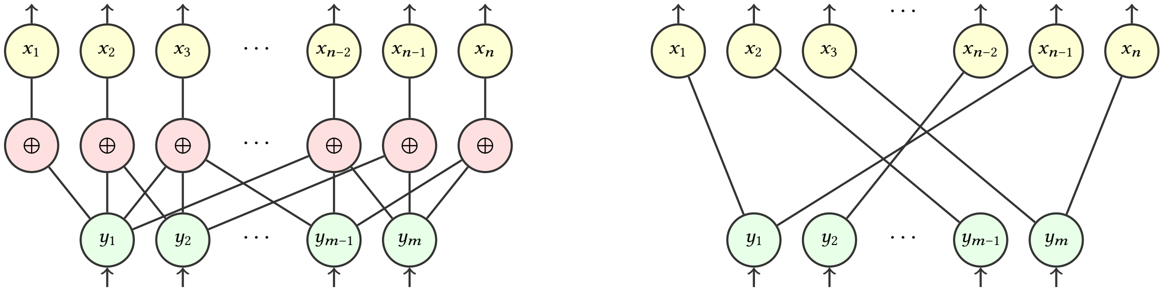
Theorem 1.5 (Weisfeiler–Leman trade-offs (informal)). *For all c, k with $1 \le c \le k-1$ and large enough n, there are pairs of graphs over n vertices that can be distinguished by k-dimensional Weisfeiler–Leman, but for which even $(k+c-1)$-dimensional Weisfeiler–Leman requires $\Omega\big(n^{k/(c+1)}\big)$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

By the well-known equivalence between Weisfeiler–Leman and fragments of first order logic with counting [17], our result also implies trade-offs between variable number and quantifier depth for such logics.

1.4 Techniques

Most of the previously known supercritical trade-offs are based on *hardness condensation* [52], that works by substituting the variables of a medium-hard problem instance with *exclusive or* (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 [12] to prove the Weisfeiler–Leman trade-offs discussed above.

The result of the recent paper [34] 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* [55]. 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



(a) Hardness condensation

(b) Variable compression

Figure 2: Hardness condensation in Figure 2a substitutes the x-variables with exclusive ors (XORs) over distinct subsets of y-variables, while variable compression in Figure 2b substitutes with y-variables directly. Note that $m \ll n$.

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 [29]. Because the correspondence is not exact, the trade-offs in [34] do not immediately transfer to proof complexity.

Using a refined graph compression and analysis, we obtain the robust Weisfeiler–Leman trade-offs stated in Theorem 1.5. Thanks to this robustness, we are able to translate these results into truly supercritical width-depth trade-offs for resolution, exporting the technique of [34] 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* is interesting in its own right and may find more applications in proof complexity.

The remaining trade-offs in this paper are obtained by applying different new *lifting theorems* to our width-depth trade-off. Lifting is a framework for transferring lower bounds from weaker computational models to comparable lower bounds for stronger models. In particular, the lifting theorems of [30, 43] 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.

1.5 Related Work

In concurrent work, Göös, Maystre, Risse and Sokolov [31] 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 in a black-box fashion 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, our results are incomparable to those in [31]. 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 [10] obtained a truly supercritical width-size trade-off for treelike resolution. We showed our treelike resolution trade-off only after the results in [10], but we use different techniques that yield much better parameters.

1.6 Organisation of This Paper

The rest of this paper is structured as follows. A proof overview is presented in Section 2, with the necessary preliminaries given along the way. We formally give our main construction in Section 3, and refer the reader to the full version of this paper [21] for complete proofs of our results. We conclude in Section 4 with some open problems.

2 Proof Overview and Preliminaries

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. Let us start by setting up some notation and giving general definitions.

For $a, b \in \mathbb{N}^+$, if $a \leq b$, we let $[a, b] = \{a, a+1, \ldots, b\}$, and we write $[1, a]$ as $[a]$. Given $k \in \mathbb{N}^+$ and $a, b \in [k]$, where $a > b$, we write $[a, b]_k = \{a, a+1, \ldots, k\} \cup \{1, 2, \ldots, b\}$. We call sets $[a, b]_k$

cyclic intervals modulo k, and sometimes omit the subscript if k is clear from the context. In this paper, $\log(\cdot)$ denotes the logarithm in base 2 and $\ln(\cdot)$ denotes the natural logarithm.

All graphs in this paper are simple, i.e., they contain no multiple edges or loops. For a graph $G = (V, E)$ and a vertex subset $W \subseteq V$, we write $G|_W$ for the subgraph induced by W, i.e., $G|_W = (V \cap W, E \cap W \times W)$. 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 a *simple* path, i.e., a path is a sequence of distinct vertices with consecutive vertices connected by an edge. The *cylinder graph with i rows and j columns* is the grid graph on $[i] \times [j]$ with the edges going from the bottom to the top in every column, i.e., the edges $\{(1, \ell), (i, \ell)\}$ for all $\ell \in [j]$, added.

2.1 The Cop-Robber Game

Let us begin with the *k-cop-robber game* [55]. Given a graph G, the cops and the robber stay on vertices of 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 moves along a path in G from his position w_1 to another vertex w_2 without visiting any vertex occupied by a cop.

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

The game ends when a cop lands at the robber's position.

We will analyze a variant of the game called the *compressed cop-robber game* [34]. It is played on a graph endowed with equivalence relations on both vertices and edges as follows.

Definition 2.1 (Graph compression [34]). Given a graph G and an equivalence relation $\equiv_V$ on $V(G)$, we say that $\equiv_V$ is *compatible* if $u \equiv_V v$ implies that u, v are non-adjacent and have the same degree. Assume that for every vertex $v \in V(G)$, we have an order of the neighbors of v. Then, 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 order of the neighbors of v_1 is the same as the position of w_2 in the order of the neighbors 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.*

Playing the cop-robber game on a graph endowed with a compressions leads to the *compressed cop-robber game*. In this game, intuitively, the cops have clones at all vertices equivalent to the vertices they occupy, and the robber must ensure that his moves are closed under $\equiv_E$ and do not visit any vertex occupied by a cop clone. The formal definition is given in Section 3.

In the specific instance we analyze, the graph is a cylinder with k rows and roughly n^k columns. We compress it by identifying vertices on the same row periodically, using a different period for every row; for details, see Section 3. Note that 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, where the graphs $\{G_n\}$ will be cylinders with k rows.

Theorem 2.2 (cop-robber). *For any parameters $k = k(n)$ and $c = c(n)$ where $1 \leq c \leq k - 1$ and $3 \leq k < n/(2 \ln n)$, there are degree-4 graphs $\{G_n\}_{n \in \mathbb{N}^+}$ 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 proof of Theorem 2.2 is deferred to the full-length version of the paper. The novelty of our analysis compared to [34] lies in having the robber play against a virtual stronger opponent, whose transition over game rounds is easier to analyze.

2.2 The Weisfeiler–Leman Algorithm

We describe the Weisfeiler–Leman algorithm on graphs; see the survey [39] for further explanations. A graph $G = (V, E, c)$ where $c \colon V \longrightarrow \mathbb{N}$ is *vertex colored.* Given $k \geq 2$ and a vertex colored graph $G = (V, E, c)$, the *k-dimensional Weisfeiler–Leman algorithm* [36, 59] iteratively refines a coloring of the k-tuples of vertices. We denote the coloring after the ith round by $\chi^{(i)} \colon V^k \longrightarrow C$, 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 $\chi^{(i-1)}(\vec{u})$:

$$\chi^{(i)}(\vec{u}) = \Big(\chi^{(i-1)}(\vec{u}), \\ \{\!\!\{ \big(\chi^{(i-1)}(\vec{u}\ [v/u_1]),\ \ldots,\ \chi^{(i-1)}(\vec{u}\ [v/u_k])\big) \mid v \in V(G) \}\!\!\} \Big).$$

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, which we present in Appendix A of the full-length version of the paper [21], Theorem 2.2 gives the following trade-off for Weisfeiler–Leman algorithms. This is a detailed version of Theorem 1.5.

Theorem 2.3 (Weisfeiler–Leman trade-offs). *For all c and k with $1 \leq c \leq k - 1$, and all n large enough, there are graph pairs over n vertices that are 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 [17, Theorem 5.2], we obtain the following trade-off between the number of variables and quantifier depth as a corollary.

Corollary 2.4. *For all c and k with $1 \le c \le k-1$, if n is large enough, there are graph pairs over n vertices that are 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., [16, 41]. 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 set 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 formula $F = D_1 \wedge \cdots \wedge D_m$ is a conjunction of clauses. The *formula width* $W(F)$ is the maximal width among 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 say that F is a *k-CNF formula* if all clauses are k-clauses. We denote by *Vars*(F) the set of variables appearing in a formula F.

The *Tseitin formula* [58] is defined for a graph G where each vertex $v \in V(G)$ has a label $\chi(v) \in \{0,1\}$, labeled 0 or 1 so that the labels sum to an odd number. The formula, denoted by $\mathrm{Ts}(G)$, has a variable x_e for every edge $e \in E(G)$ and is defined to be the CNF formula 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, i.e.,

$$\sum_{e \ni v} x_e = \chi(v). \tag{1}$$

The formula $\mathrm{Ts}(G)$ is unsatisfiable due to the handshaking lemma.

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*

$$\frac{C \vee x \qquad D \vee \overline{x}}{C \vee D}. \tag{2}$$

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. A refutation π is *treelike* if G_π is a tree.

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 a largest clause in π. 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

Our first proof complexity result is the following truly supercritical width-depth trade-off for resolution.

Theorem 2.5 (Width-depth trade-offs). *For all n and integer parameters $k = k(n)$, $c = c(n)$ where $3 \le c \le k-1 < \frac{n}{2\ln n}$, there is a 4-CNF formula F over $N \in [2k^2 n^{c+1}, 40k^2(2n)^{c+1}]$ variables, having formula size $O(N)$, which resolution can refute in width $k+3$ and size $O(k^2(4n)^k)$ simultaneously, but for which any refutation of width at most $k+c$ must have depth at least $\Omega(n^k)$.*

The formula F above is a Tseitin formula after a variable identification obtained from a graph compression, which we formally demonstrate in Definition 3.4. The theorem follows from Theorem 2.2 by applying somewhat standard translations between the (compressed) cop-robber games and resolution. The proof details can be found in the full-length version of the paper.

The two examples in Theorem 1.4 follow from Theorem 2.5 by taking $k(n) = \lfloor n/(2\log n)\rfloor$ and, for item (1), setting $c(n)$ to be a large constant, or, for item (2), setting $c(n)$ to be $\lfloor \frac{1+\delta}{2}k \rfloor$.

2.5 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: $\mathrm{XOR}_m : \{0,1\}^m \longrightarrow \{0,1\}$, defined as $\mathrm{XOR}_m(x_1, \ldots, x_m) = \bigoplus_{i \in [m]} x_i$, and $\mathrm{IND}_m : [m] \times \{0,1\}^m \longrightarrow \{0,1\}$, 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:

$$y_{4,1} \vee y_{4,2} \vee y_{5,1} \vee \overline{y}_{5,2}\,, \qquad y_{4,1} \vee y_{4,2} \vee \overline{y}_{5,1} \vee y_{5,2}\,, \tag{3}$$

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

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 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 2.6 (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 leave the proof of this theorem the full-length version of the paper. We can now apply this theorem to our width-depth trade-off to obtain the supercritical trade-offs for treelike resolution.

Theorem 2.7 (Width-size trade-offs). *For all n and integer parameters $m(n) \ge 3$, $k(n) \in [4, \frac{n}{2\ln n}]$, and $\varepsilon(n) \in (\frac{4}{k}, 1-\frac{1}{k})$, there are $4m$-CNF formulas F_N over $N \in [2k^2 n^{\lfloor \varepsilon k \rfloor} m, 40k^2(2n)^{\lfloor \varepsilon k \rfloor} m]$ variables, having formula size $O(16^m \cdot N)$, which resolution can refute in width $m(k+3)$, but for which any treelike refutation of width at most $(m-1)(1+\varepsilon)k$ must have size at least $2^{\Omega(n^k)}$.*

Proof. Let F be the 4-CNF formulas from Theorem 2.5 with parameter $c = \lfloor \varepsilon k \rfloor - 1 \in [3, k-1]$, and define $F_N = \mathrm{XOR}_m(F)$. Then $S(F_N) = \mathrm{O}(2^{4(m-1)} \cdot 4m \cdot S(F)) = \mathrm{O}(2^{4m} \cdot m \cdot |Vars(F)|) = \mathrm{O}(16^m \cdot N)$, 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 2.6, there is a refutation of F in width $(1+\varepsilon)k$ and depth $\log s$. The theorem follows since Theorem 2.5 implies that $\log s = \Omega(n^k)$. □

We obtain Theorem 1.3 follows from Theorem 2.7 by setting $k(n) = \lfloor n/(2\log n) \rfloor$, and choosing the remaining parameters to be $m = 256$ and $\varepsilon = 0.41$ for item (1), and $m = \lfloor \sqrt{n} \rfloor$ and $\varepsilon = \frac{100}{\sqrt{n}}$ for item (2), respectively.

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 [6].[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

$$(x_{i,1} \to y_{i,1}) \wedge \ldots \wedge (x_{i,m} \to y_{i,m}), \tag{5}$$

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 $\leq m$ clauses. Note that the width of $\mathrm{IND}_m(F)$ is $2W(F)$ and the number of clauses is $|\mathrm{IND}_m(F)| \leq m^{W(F)}|F| + nm$. Using this gadget, we obtain our lifting theorem for resolution.

Theorem 2.8 (Lifting for resolution). *For any $m, n \geq 1$ and a CNF formula F over n variables, 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.*

The theorem holds for any gadget size, and the size-width relation it provides is nearly tight (see Lemma 2.9 below). The proof, included in the full-length version of the paper [21], is simple and 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 2.9. *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 $\mathrm{O}(s \cdot m^{w+1})$.*

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 $\mathrm{O}(nm)$ 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 $m^{w'+1} + m = \mathrm{O}(m^{w+1})$ steps per new clause in Π. □

We can now apply Theorem 2.8 to our width-depth trade-off to obtain supercritical size-depth trade-offs for resolution.

Theorem 2.10 (Resolution size-depth trade-offs). *For all n and integer parameters $m(n)$, $k(n)$, and $c(n)$ where $3 \leq c \leq k-1 < \frac{n}{2\ln n}$, there are 8-CNF formulas F_N over $N = O(mk^2(2n)^{c+1})$ variables, having formula size $S(F_N) = \mathrm{O}(m^4 k^2 (2n)^{c+1})$, which resolution can refute in size $\mathrm{O}(m^{k+4} k^2 (4n)^k)$, but for which any refutation of size at most $\left(\frac{m+1}{2}\right)^{k+c}$ must have depth at least $\Omega(n^k)$.*

Proof. Let $F_N = \mathrm{IND}_m(F)$, where F is the formula obtained from the width-depth trade-off in Theorem 2.5 for parameters c, k and n. Note that F_N is a 8-CNF formula of size $\mathrm{O}(m^4 k^2 (2n)^{c+1})$. Since by Theorem 2.5 F has a resolution refutation of width $k+3$ and size $O(k^2(4n)^k)$, we have by Lemma 2.9 that $\mathrm{IND}_m(F)$ has a resolution refutation of size $\mathrm{O}(m^{k+4} k^2 (4n)^k)$. The lower bounds follows from combining the lifting result in Theorem 2.8 with the width-depth trade-off in Theorem 2.5. □

2.6 Trade-offs for 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* [30], which are a generalisation of rectangle-DAGs introduced in [51] and simplified in [47, 56].

Definition 2.11 (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

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

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

$$(z, i) \in \mathsf{Search}(F) \iff C_i(z) = 0, \tag{6}$$

where we view the clause C_i as a circuit evaluated on $z \in \{0, 1\}^n$. We are sometimes interested in making the input domain bipartite. For that purpoes, we partition the variables of F into two parts, view $\{0, 1\}^n$ as a product $X \times Y$ accordingly, and define the relation $\mathsf{Search}^{X,Y}(F) \subseteq (X \times Y) \times [m]$ by $((x, y), i) \in \mathsf{Search}(F) \iff C_i((x, y)) = 0$. 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* [38] $\mathsf{mKW}(f) \subseteq (f^{-1}(1) \times f^{-1}(0)) \times [n]$ is defined as

$$((x, y), i) \in \mathsf{mKW}(f) \iff x_i > y_i\,. \tag{7}$$

The DAG-like version of the monotone Karchmer–Wigderson relation [47, 51, 56] 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 [28, 50], there is a way of translating between the composed search problems; see e.g. [30] for a proof.

Fact 2.12. *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) $\mathsf{Search}(F) \circ \mathrm{IND}^n_m$ reduces to $\mathsf{mKW}(f)$. In particular, an $\mathcal{F}$-DAG solving $\mathsf{mKW}(f)$ implies an $\mathcal{F}$-DAG solving $\mathsf{Search}(F) \circ \mathrm{IND}^n_m$ 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 2.13 (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 $\mathsf{Search}(F) \circ \mathrm{IND}^n_m$, then F has a resolution refutation of width w and depth dw.*

The proof of this theorem can be found in the full version of the paper [21].

Combining this lifting theorem with our width-depth trade-off for resolution in Theorem 2.5, we obtain the supercritical size-depth trade-offs for monotone (real) circuits.

Theorem 2.14 (Monotone circuit trade-offs). *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 2.5, 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 Theorem 2.12 to F. We have that $N = \mathrm{O}(m^4 k^2 (2n)^{c+1})$. Since by Theorem 2.5, F has a resolution refutation of width $k + 3$ and size $O(k^2 (4n)^k)$, we have by Lemma 2.9 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 Theorem 2.12 $\mathsf{mKW}(g_N)$ 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 Theorem 2.12 this implies there is a triangle-DAG solving $\mathsf{Search}(F) \circ \mathrm{IND}^n_m$ in the same size and depth. Finally, combining the triangle-DAG lifting theorem (Theorem 2.13) and the width-depth trade-off (Theorem 2.5) we conclude that if $s \leq \frac{1}{2}m^{(1-\delta)(k+c)}$ then $d = \Omega(n^k/(k + c)) = \Omega(n^k/k)$. □

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

Theorem 2.15 (Cutting planes trade-offs). *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 $O(m^{k+4} k^2 (4n)^k)$, 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 2.14, by applying the lifting theorem (Theorem 2.13) to the width-depth trade-off (Theorem 2.5) together with Theorem 2.12, and using Lemma 2.9 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 $(\bigwedge_{\ell \in [w]} x_{i_\ell, j_\ell}) \rightarrow x_{C,J}$, a 2-clause $x_{C,J} \rightarrow y_{C,J}$, and a 3-CNF encoding of $y_{C,J} \rightarrow \bigvee_{\ell \in [w]} y_{i_\ell, j_\ell}^{\beta_\ell}$. 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 $\mathrm{Search}^{X,Y}(\mathrm{IND}_m(F))$ reduces to $\mathrm{Search}^{\widetilde{X},\widetilde{Y}}(\widetilde{\mathrm{IND}}_m(F))$, 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 $\phi_X : \{0,1\}^X \rightarrow \{0,1\}^{\widetilde{X}}$ and $\phi_Y : \{0,1\}^Y \rightarrow \{0,1\}^{\widetilde{Y}}$ 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 O be the set of possible outputs of $\mathrm{Search}^{X,Y}(\mathrm{IND}_m(F))$, which we view as the set of clauses of $\mathrm{IND}_m(F)$. Similarly, let $\widetilde{O}$ be the set of clauses of $\widetilde{\mathrm{IND}}_m(F)$. We can define an injective map $\phi_{\widetilde{O}} : \widetilde{O} \rightarrow O$ 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 $\mathrm{Search}^{\widetilde{X},\widetilde{Y}}(\widetilde{\mathrm{IND}}_m(F))$, we can create an $\mathcal{F}$-DAG Γ solving $\mathrm{Search}^{X,Y}(\mathrm{IND}_m(F))$ 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 $\mathrm{Search}^{X,Y}(\mathrm{IND}_m(F))$.

Proof of Theorem 2.15. Let $F_N = \widetilde{\mathrm{IND}}_m(F)$, where F is the formula obtained from Theorem 2.5, 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 2.5, F has a resolution refutation of width $k+3$ and size $O(k^2(4n)^k)$, we have by Lemma 2.9 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, there is a triangle-DAG solving $\mathrm{Search}^{X,Y}(\mathrm{IND}_m(F))$ of size s and depth d, using the fact above that $\mathrm{Search}^{X,Y}(\mathrm{IND}_m(F))$ reduces to $\mathrm{Search}^{\widetilde{X},\widetilde{Y}}(\widetilde{\mathrm{IND}}_m(F))$. By Theorem 2.12 this gives a triangle-DAG solving $\mathrm{Search}(F) \circ \mathrm{IND}_m^n$ of the same size and depth. Finally, combining the triangle-DAG lifting theorem (Theorem 2.13) and the width-depth trade-off (Theorem 2.5), we conclude that if $s \leq \frac{1}{2} m^{(1-\delta)(k+c)}$ then $d = \Omega(n^k/(k+c)) = \Omega(n^k/(k+c))$. □

Let us verify that Theorem 1.1 follows from Theorem 2.14, and Theorem 1.2 from Theorem 2.15. Indeed, to obtain item (1), we can set k to be a sufficiently large constant, and the remaining parameters to be $c = \lfloor 0.41k \rfloor$, $w = k + c$, $\delta = \frac{1}{200}$, and $m = n^{500}$. For item (2), we choose $k = \lfloor \frac{n}{4\log n} \rfloor$, and let $c = \lfloor \sqrt{k} \rfloor$, $w = k + c$, $\delta = \frac{1}{4\sqrt{k}}$, and $m = \lfloor n^{3/\delta} \rfloor$, i.e., $m = \lfloor n^{12\sqrt{k}} \rfloor$.

3 Formal Construction of Compressed Cylinder, Cop-Robber Game, and Tseitin Formula

In this section, we give the formal definitions of the compressed cop-robber game, the specific family of graph compressions we use to establish the lower bound on the number of rounds in Theorem 2.2, as well as the family of formulas exhibiting the width-depth trade-off in Theorem 2.5. We start by specifying how the cop-robber game is played when the graph is endowed with a graph compression as in Definition 2.1.

Definition 3.1 (Compressed cop-robber game [34]). Given a graph compression $(G, \equiv_V, \equiv_E)$, the *compressed k-cop-robber game* on G proceeds as follows. The cops and the robber stay on vertices of G 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 of the game, the following happens in turn:

(1) If there is no lifted cop, the cops choose and lift one. Then, a lifted cop signals a vertex v to the robber.

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

  (a) (M is closed under $\equiv_E$.) For two equivalent edges $e \equiv_E e'$, it holds that $e \in M$ if and only if $e' \in M$.

  (b) (No vertex equivalence class touched by M is occupied.) Denote the set of vertices occupied by the cops after (1) by $\mathbb{C} \subseteq V$. No edge in M is incident to a vertex in $\mathbb{C}$, i.e., $\mathbb{C} \cap V(M) = \emptyset$.

  (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 $w_1 \neq w_2$.)

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

The game ends when a cop is at a vertex equivalent to the vertex occupied by the robber.

As mentioned earlier, we analyze a specific family of graph compressions, which we call *compressed cylinders*.

Definition 3.2 (Compressed cylinder). Given integers k, c, n such that $k \geq 3$, $c \in [1, k-1]$, and n is sufficiently large, we let

$(G_{\mathrm{cyl}}, \equiv_V, \equiv_E)$ be the following graph compression. The graph G_{cyl} is a cylinder with k rows and $L + 2r$ columns. (The parameters L and r are specified below in item 3.)

(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 $E(G_{\mathrm{cyl}})$ via Definition 2.1. Here, we assume the edges incident to a vertex are ordered by the canonical choice (left, right, up, down), adjusted to a subset of size 3 for vertices on boundary columns.

(3) *Choice of parameters.* We set m_i, L, and r in G_{cyl} as follows. Take k pairwise coprime numbers $P_1, \ldots, P_k$ in $[n, 2n]$. (This is possible by the prime number theorem if $n > 2k \ln k + C$ holds for some absolute constant C, and we picked n to be large enough.) Then for each $i \in [k]$, we let:

$$m_i = 2(k + c) \cdot P_i \cdots P_{i+c}\,, \tag{8}$$

$$L = 2(k + c) \cdot P_1 \cdots P_k\,, \tag{9}$$

$$r = k + c + 1\,. \tag{10}$$

This definition extends the construction due to Grohe et al. [34] by letting the row periods depend on the parameter c. Their original construction is recovered in the case when $c = 2$.

We are now ready to state Theorem 2.2 with full details.

Theorem 3.3 (Cop-robber (detailed)). *For any $k \geq 3$, $c \leq k - 1$, and large enough n, it holds that $k + 1$ cops can win the compressed cop-robber game on $(G_{\mathrm{cyl}}, \equiv_V, \equiv_E)$, but as long as there are at most $k + c$ cops, the robber can survive for $(L - 2r)/(8(k + c))$ rounds.*

Turning our attention to CNF formulas, we define the *compressed Tseitin formulas*, for which we prove our width-depth trade-off. Given a graph compression, we denote by $/_{\equiv_V}$ the map from the vertices to the vertex equivalence classes, that maps every vertex to its equivalence class, and similarly by $/_{\equiv_E}$ the map from the edges to their equivalence classes.

Definition 3.4 (Compressed Tseitin Formula). Given a Tseitin formula $\mathrm{Ts}(G)$ 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 edge equivalence class, introduce a single, new variable $x_{e/\equiv_E}$. Replace every occurrence of a variable x_e in $\mathrm{Ts}(G)$ with the variable $x_{e/\equiv_E}$. We call the resulting CNF formula the *compressed Tseitin formula on* $(G, \equiv_V, \equiv_E)$, and denote it by $\mathrm{Ts}(G)_\equiv$.

Consider the Tseitin formula on G_{cyl} with the labelling $\chi(v)$ giving label 1 only to the first vertex in the first row, i.e., $\chi(v) = 1$ if and only if $v = (1, 1)$. Together with the graph compression in Definition 3.2, this yields a compressed Tseitin formula $\mathrm{Ts}(G_{\mathrm{cyl}})_\equiv$ as in Definition 3.4. We show our width-depth trade-off in Theorem 2.5 for the formulas obtained from this construction.

The proof of the trade-off exploits a connection between the compressed Tseitin formula and the compressed cop-robber game played on the same graph compression. Indeed, the upper bound in Theorem 2.5 is a resolution refutation corresponding to the simple strategy for $k + 1$ cops in the compressed game.

Lemma 3.5 (Small-width refutation). *The formula $\mathrm{Ts}(G_{\mathrm{cyl}})_\equiv$ has a resolution refutation with width $k + 3$ and size $O((L + r)2^k k)$.*

Furthermore, the cops can win the compressed game by playing according to a strategy derived from a refutation of the compressed formula $\mathrm{Ts}(G_{\mathrm{cyl}})_\equiv$, as stated below.

Lemma 3.6 (Cops simulate refutation). *If there is a width-w and depth-d resolution refutation of $\mathrm{Ts}(G_{\mathrm{cyl}})_\equiv$, then in the compressed $(w + 1)$-cop-robber game on G_{cyl}, where the robber starts the game at the vertex (1,1), the cops can win in $d + 1$ rounds.*

Taken together, Theorem 3.3 and Lemma 3.6 imply the depth lower bound for bounded width refutations in Theorem 2.5. The proofs of Theorem 3.3 and Lemma 3.5, as well as that of a more general version of Lemma 3.6 that applies to any graph compression, can be found in the full-length version of the paper [21].

4 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.

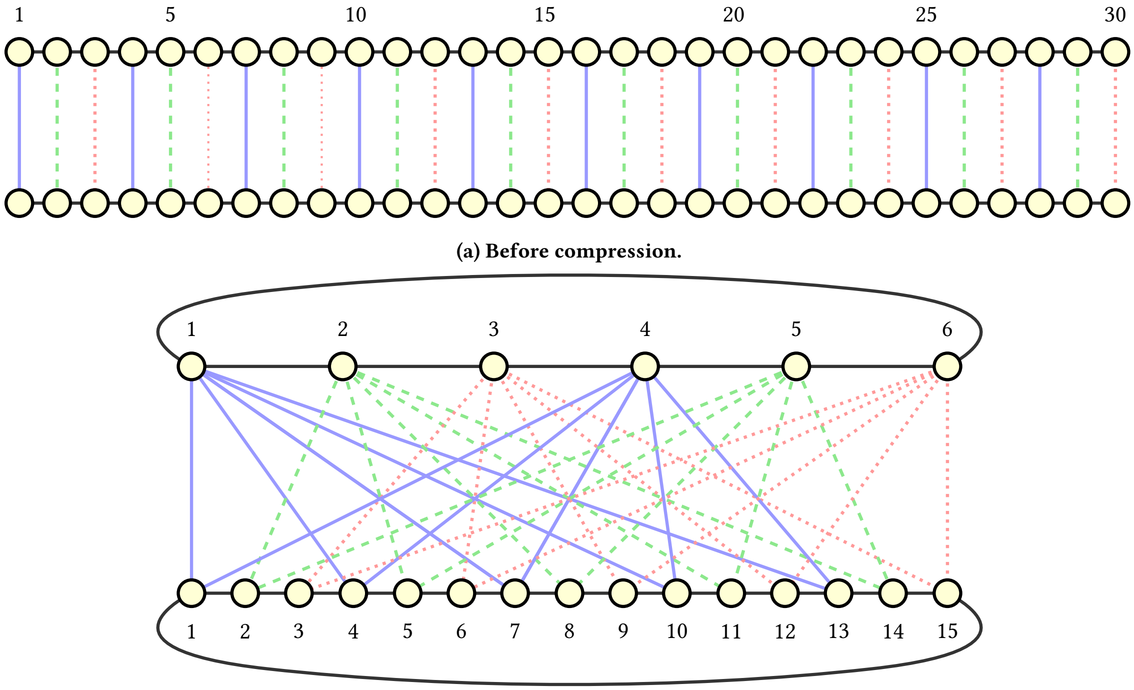
Supercritical Trade-offs for Perfect Matching and Tseitin. Having established truly supercritical trade-offs for monotone circuits and cutting planes, we find it natural to ask for more examples of this phenomenon. As mentioned in the introduction, it is possible that the perfect matching problem exhibits 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* [5] could be efficiently translated into cutting planes. In fact, as was shown in [23], 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

The authors wish to thank Christoph Berkholz, Jonas Conneryd, Daniel Neuen, and Alexander Razborov for helpful discussions.



(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 line style.

We would also like to thank the participants of the Oberwolfach workshop *Proof Complexity and Beyond* in March 2024 and of the Dagstuhl workshop 24421 *SAT and Interactions* for their feedback.

Susanna F. de Rezende received funding from the Knut and Alice Wallenberg grant KAW 2021.0307, ELLIIT, and the Swedish Research Council grant 2021-05104. Noah Fleming was funded by NSERC. Duri Andrea Janett and Jakob Nordström received funding from the Independent Research Fund Denmark grant 9040-00389B, and Jakob Nordström was also supported by the Swedish Research Council grant 2016-00782. Shuo Pang was funded by the European Union MSCA Postdoctoral Fellowships 2023 project 101146273 NoShortProof. Views expressed are the authors' and do not reflect the European Union or the Research Executive Agency.

References

[1] Noga Alon and Ravi B. Boppana. 1987. The Monotone Circuit Complexity of Boolean Functions. *Combinatorica* 7, 1 (March 1987), 1–22. doi:10.1007/BF02579196

[2] Alexander E. Andreev. 1985. On a Method for Obtaining Lower Bounds for the Complexity of Individual Monotone Functions. *Soviet Mathematics Doklady* 31, 3 (1985), 530–534. English translation of a paper in *Doklady Akademii Nauk SSSR*.

[3] László Babai. 2016. Graph Isomorphism in Quasipolynomial Time [Extended Abstract]. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC '16)*. 684–697. doi:10.1145/2897518.2897542

[4] Paul Beame, Chris Beck, and Russell Impagliazzo. 2016. Time-Space Tradeoffs in Resolution: Superpolynomial Lower Bounds for Superlinear Space. *SIAM J. Comput.* 45, 4 (August 2016), 1612–1645. doi:10.1137/130914085 Preliminary version in *STOC '12*.

[5] Paul Beame, Noah Fleming, Russell Impagliazzo, Antonina Kolokolova, Denis Pankratov, Toniann Pitassi, and Robert Robere. 2018. Stabbing Planes. In *Proceedings of the 9th Innovations in Theoretical Computer Science Conference (ITCS '18) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 94)*. 10:1–10:20. doi:10.4230/LIPIcs.ITCS.2018.10

[6] Paul Beame, Trinh Huynh, and Toniann Pitassi. 2010. Hardness Amplification in Proof Complexity. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC '10)*. 87–96. doi:10.1145/1806689.1806703

[7] Paul Beame and Toniann Pitassi. 1998. Propositional Proof Complexity: Past, Present, and Future. *Bulletin of the European Association for Theoretical Computer Science* 65 (June 1998), 66–89.

[8] Chris Beck, Jakob Nordström, and Bangsheng Tang. 2013. Some Trade-off Results for Polynomial Calculus. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC '13)*. 813–822. doi:10.1145/2488608.2488711

[9] Christoph Berkholz. 2012. On the Complexity of Finding Narrow Proofs. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '12)*. 351–360. doi:10.1109/FOCS.2012.48

[10] Christoph Berkholz, Moritz Lichter, and Harry Vinall-Smeeth. 2024. Supercritical Size-Width Tree-Like Resolution Trade-Offs for Graph Isomorphism. *arXiv preprint arXiv 2407.17947* (July 2024). doi:10.48550/arXiv.2407.17947 arXiv:2407.17947 [cs.LO]

[11] Christoph Berkholz and Jakob Nordström. 2020. Supercritical Space-Width Trade-offs for Resolution. *SIAM J. Comput.* 49, 1 (February 2020), 98–118. doi:10.1137/16M1109072 Preliminary version in *ICALP '16*.

[12] Christoph Berkholz and Jakob Nordström. 2023. Near-Optimal Lower Bounds on Quantifier Depth and Weisfeiler-Leman Refinement Steps. *J. ACM* 70, 5 (October 2023), 32:1–32:32. doi:10.1145/3195257 Preliminary version in *LICS '16*.

[13] María Bonet, Toniann Pitassi, and Ran Raz. 1997. Lower Bounds for Cutting Planes Proofs with Small Coefficients. *Journal of Symbolic Logic* 62, 3 (September 1997), 708–728. doi:10.2307/2275569 Preliminary version in *STOC '95*.

[14] Joshua Buresh-Oppenheim, Nicola Galesi, Shlomo Hoory, Avner Magen, and Toniann Pitassi. 2006. Rank Bounds and Integrality Gaps for Cutting Planes Procedures. *Theory of Computing* 2, 4 (2006), 65–90. doi:10.4086/toc.2006.v002a004 Preliminary version in *FOCS '03*.

[15] Sam Buss and Neil Thapen. 2024. *A Simple Supercritical Tradeoff between Size and Height in Resolution.* Technical Report TR24-001. Electronic Colloquium on Computational Complexity (ECCC). https://eccc.weizmann.ac.il/report/2024/001/

[16] Samuel R. Buss and Jakob Nordström. 2021. Proof Complexity and SAT Solving. In *Handbook of Satisfiability* (2nd ed.), Armin Biere, Marijn J. H. Heule,

Hans van Maaren, and Toby Walsh (Eds.). Frontiers in Artificial Intelligence and Applications, Vol. 336. IOS Press, Chapter 7, 233–350. doi:10.3233/FAIA200990

[17] Jin-yi Cai, Martin Fürer, and Neil Immerman. 1992. An Optimal Lower Bound on the Number of Variables for Graph Identifications. *Combinatorica* 12, 4 (1992), 389–410. doi:10.1007/BF01305232 Preliminary version in *FOCS '89*.

[18] Arkadev Chattopadhyay and Pavel Dvořák. 2024. *Super-critical Trade-offs in Resolution over Parities Via Lifting.* Technical Report TR24-132. Electronic Colloquium on Computational Complexity (ECCC). https://eccc.weizmann.ac.il/report/2024/132/

[19] William Cook, Collette Rene Coullard, and György Turán. 1987. On the Complexity of Cutting-Plane Proofs. *Discrete Applied Mathematics* 18, 1 (November 1987), 25–38. doi:10.1016/0166-218X(87)90039-4

[20] Daniel Dadush and Samarth Tiwari. 2020. On the Complexity of Branching Proofs. In *Proceedings of the 35th Annual Computational Complexity Conference (CCC '20) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 169).* 34:1–34:35. doi:10.4230/LIPIcs.CCC.2020.34

[21] Susanna F. de Rezende, Noah Fleming, Duri Andrea Janett, Jakob Nordström, and Shuo Pang. 2024. *Truly Supercritical Trade-offs for Resolution, Cutting Planes, Monotone Circuits, and Weisfeiler–Leman.* Technical Report TR24-185. Electronic Colloquium on Computational Complexity (ECCC). https://eccc.weizmann.ac.il/report/2024/185/

[22] Jack Edmonds. 1965. Paths, Trees, and Flowers. *Canadian Journal of Mathematics* 17 (1965), 449–467. doi:10.1007/978-0-8176-4842-8_26

[23] Noah Fleming, Mika Göös, Russell Impagliazzo, Toniann Pitassi, Robert Robere, Li-Yang Tan, and Avi Wigderson. 2021. On the Power and Limitations of Branch and Cut. In *Proceedings of the 36th Annual Computational Complexity Conference (CCC '21) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 200).* 6:1–6:30. doi:10.4230/LIPIcs.CCC.2021.6

[24] Noah Fleming, Denis Pankratov, Toniann Pitassi, and Robert Robere. 2022. Random $\Theta(\log n)$-CNFs are Hard for Cutting Planes. *J. ACM* 69, 3 (June 2022), 19:1–19:32. doi:10.1145/3486680 Preliminary version in *FOCS '17*.

[25] Noah Fleming and Toniann Pitassi. 2022. Reflections on Proof Complexity and Counting Principles. In *Alasdair Urquhart on Nonclassical and Algebraic Logic and Complexity of Proofs*, Ivo Dünsch and Edwin Mares (Eds.). Outstanding Contributions to Logic, Vol. 22. Springer, Chapter 18, 497–520. doi:10.1007/978-3-030-71430-7_18

[26] Noah Fleming, Toniann Pitassi, and Robert Robere. 2022. Extremely deep proofs. In *Proceedings of the 13th Innovations in Theoretical Computer Science Conference (ITCS '22)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 70:1–70:23. doi:10.4230/LIPIcs.ITCS.2022.70

[27] Martin Fürer. 2001. 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) (Lecture Notes in Computer Science, Vol. 2076)*. Springer, 322–333. doi:10.1007/3-540-48224-5_27

[28] Anna Gál. 2001. A Characterization of Span Program Size and Improved Lower Bounds for Monotone Span Programs. *Computational Complexity* 10, 4 (December 2001), 277–296. doi:10.1007/s000370100001 Preliminary version in *STOC '98*.

[29] Nicola Galesi, Navid Talebanfard, and Jacobo Torán. 2020. Cops-robber games and the resolution of Tseitin formulas. *ACM Transactions on Computation Theory (TOCT)* 12, 2 (May 2020), 1–22. doi:10.1145/3378667 Preliminary version in *SAT '18*.

[30] Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. 2020. Monotone Circuit Lower Bounds from Resolution. *Theory of Computing* 16, 13 (2020), 1–30. doi:10.4086/toc.2020.v016a013 Preliminary version in *STOC '18*.

[31] Mika Göös, Gilbert Maystre, Kilian Risse, and Dmitry Sokolov. 2024. *Super-critical Tradeoffs for Monotone Circuits.* Technical Report TR24-186. Electronic Colloquium on Computational Complexity (ECCC). https://eccc.weizmann.ac.il/report/2024/186/ To appear in *STOC '25*.

[32] Martin Grohe. 2021. The Logic of Graph Neural Networks. In *Proceedings of the 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '21)*. 1–17. doi:10.1109/LICS52264.2021.9470677

[33] Martin Grohe, Moritz Lichter, and Daniel Neuen. 2023. The Iteration Number of the Weisfeiler-Leman Algorithm. In *Proceedings of the 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '23)*. 1–13. doi:10.1109/LICS56636.2023.10175741

[34] Martin Grohe, Moritz Lichter, Daniel Neuen, and Pascal Schweitzer. 2023. 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)*. 798–809. doi:10.1109/FOCS57990.2023.00052

[35] Pavel Hrubeš and Pavel Pudlák. 2017. Random Formulas, Monotone Circuits, and Interpolation. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS '17)*. 121–131. doi:10.1109/FOCS.2017.20

[36] Neil Immerman and Eric Lander. 1990. Describing Graphs: A First-Order Approach to Graph Canonization. In *Complexity Theory Retrospective: In Honor of Juris Hartmanis on the Occasion of His Sixtieth Birthday, July 5, 1988*, Alan L. Selman (Ed.). Springer, 59–81. doi:10.1007/978-1-4612-4478-3_5

[37] Stasys Jukna. 2012. *Boolean Function Complexity - Advances and Frontiers.* Algorithms and combinatorics, Vol. 27. Springer. doi:10.1007/978-3-642-24508-4

[38] Mauricio Karchmer and Avi Wigderson. 1990. Monotone Circuits for Connectivity Require Super-Logarithmic Depth. *SIAM Journal on Discrete Mathematics* 3, 2 (1990), 255–265. doi:10.1137/0403021 Preliminary version in *STOC '88*.

[39] Sandra Kiefer. 2020. The Weisfeiler-Leman algorithm: an exploration of its power. *ACM SIGLOG News* 7, 3 (July 2020), 5–27. doi:10.1145/3436980.3436982

[40] Sandra Kiefer and Pascal Schweitzer. 2019. Upper bounds on the quantifier depth for graph differentiation in first order logic. *Logical Methods in Computer Science* 15, 2, Article 19 (May 2019), 19:1–19:19 pages. doi:10.23638/LMCS-15(2:19)2019 Preliminary version in *LICS '16*.

[41] Jan Krajíček. 2019. *Proof Complexity.* Encyclopedia of Mathematics and Its Applications, Vol. 170. Cambridge University Press. doi:10.1017/9781108242066

[42] Moritz Lichter, Ilia Ponomarenko, and Pascal Schweitzer. 2019. 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)*. 1–13. doi:10.1109/lics.2019.8785694

[43] Shachar Lovett, Raghu Meka, Ian Mertz, Toniann Pitassi, and Jiapeng Zhang. 2022. Lifting with Sunflowers. In *Proceedings of the 13th Innovations in Theoretical Computer Science Conference (ITCS '22) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 215)*. 104:1–104:24. doi:10.4230/LIPIcs.ITCS.2022.104

[44] Christopher Morris, Yaron Lipman, Haggai Maron, Bastian Rieck, Nils M. Kriege, Martin Grohe, Matthias Fey, and Karsten Borgwardt. 2023. Weisfeiler and Leman go Machine Learning: The Story so far. *Journal of Machine Learning Research* 24, 333 (2023), 1–59. http://jmlr.org/papers/v24/22-0240.html

[45] Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. 2019. Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI '19)*. AAAI Press, 4602–4609. doi:10.1609/aaai.v33i01.33014602

[46] Pavel Pudlák. 1997. Lower Bounds for Resolution and Cutting Plane Proofs and Monotone Computations. *Journal of Symbolic Logic* 62, 3 (September 1997), 981–998. doi:10.2307/2275583

[47] Pavel Pudlák. 2010. 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) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 8)*. 30–41. doi:10.4230/LIPIcs.FSTTCS.2010.30

[48] Ran Raz and Avi Wigderson. 1992. Monotone circuits for matching require linear depth. *J. ACM* 39, 3 (July 1992), 736–744. doi:10.1145/146637.146684

[49] Alexander A. Razborov. 1985. Lower Bounds for the Monotone Complexity of Some Boolean Functions. *Soviet Mathematics Doklady* 31, 2 (1985), 354–357. English translation of a paper in *Doklady Akademii Nauk SSSR*.

[50] Alexander A. Razborov. 1990. Applications of Matrix Methods to the Theory of Lower Bounds in Computational Complexity. *Combinatorica* 10, 1 (March 1990), 81–93. doi:10.1007/BF02122698

[51] Alexander A. Razborov. 1995. Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izvestiya: Mathematics* 59 (February 1995), 205–227. doi:10.1070/IM1995v059n01ABEH000009

[52] Alexander A. Razborov. 2016. A New Kind of Tradeoffs in Propositional Proof Complexity. *J. ACM* 63, 2, Article 16 (April 2016), 16:1–16:14 pages. doi:10.1145/2858790

[53] Alexander A. Razborov. 2017. On the Width of Semialgebraic Proofs and Algorithms. *Mathematics of Operations Research* 42, 4 (May 2017), 1106–1134. doi:10.1287/moor.2016.0840

[54] Alexander A. Razborov. 2018. On Space and Depth in Resolution. *Computational Complexity* 27, 3 (September 2018), 511–559. doi:10.1007/s00037-017-0163-1

[55] Paul D Seymour and Robin Thomas. 1993. Graph searching and a min-max theorem for tree-width. *Journal of Combinatorial Theory, Series B* 58, 1 (May 1993), 22–33. doi:10.1006/jctb.1993.1027

[56] Dmitry Sokolov. 2017. Dag-Like Communication and Its Applications. In *Proceedings of the 12th International Computer Science Symposium in Russia (CSR '17) (Lecture Notes in Computer Science, Vol. 10304)*. Springer, 294–307. doi:10.1007/978-3-319-58747-9_26

[57] Dmitry Sokolov. 2024. Random $(\log n)$-CNF Are Hard for Cutting Planes (Again). In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24)*. 2008–2015. doi:10.1145/3618260.3649636

[58] Grigori Tseitin. 1968. On the Complexity of Derivation in Propositional Calculus. In *Structures in Constructive Mathematics and Mathematical Logic, Part II*, A. O. Silenko (Ed.). Consultants Bureau, New York-London, 115–125. doi:10.1007/978-3-642-81955-1_28

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

Received 2024-11-04; accepted 2025-02-01