UNIVERSITY OF COPENHAGEN
DEPARTMENT OF COMPUTER SCIENCE



Ph.D. Thesis

Duri Andrea Janett

Trade-offs and Lower Bounds for Proofs, Circuits, and Weisfeiler–Leman

Advisor: Prof. Jakob Nordström

This thesis has been submitted to the Ph.D. School of The Faculty of Science, University of Copenhagen on August 31, 2026.

Abstract

This thesis is a synopsis of three papers focussing on propositional proof complexity, which studies efficient certificates of unsatisfiability. Our results include trade-offs between computational resources for proof systems, circuits, and the Weisfeiler–Leman algorithm, as well as size lower bounds for proofs and corresponding communication models.

Paper A establishes so-called *supercritical* trade-offs for resolution, cutting planes, monotone circuits, and Weisfeiler–Leman. Notably, our results for proofs are supercritical even when measured in terms of the instance size, not just in terms of the number of variables in the instance.

In Paper B, we investigate a succinct encoding of the clique formulas, expressing the contradictory claim that a given graph with no clique of size k contains a k-clique. For randomly sampled dense graphs, we establish exponential size lower bounds for cutting planes and bounded-depth resolution over parities. We also show that finding a falsified clause in these formulas requires polynomial randomized communication complexity.

Lastly, in Paper C, we introduce a problem for which proving lower bounds may be viewed as an intermediate step between proving lower bounds for random $\Theta(\log n)$-CNF formulas and proving lower bounds for Nisan–Wigderson-style pseudorandom generator tautologies in the cutting planes proof system. We prove exponential lower bounds on the size of rectangle-DAGs solving this problem. For this, we develop a new technique based on the approximation method.

Resumé

Denne afhandling er en synopsis af tre artikler inden for propositional beviskompleksitet, som fokuserer på effektive certifikater for uopfyldelighed, samt beslægtede områder. Vores resultater omfatter afvejninger mellem beregningsressourcer for bevis-systemer, kredsløb og Weisfeiler–Leman-algoritmen, samt nedre grænser for størrelsen af beviser og tilsvarende kommunikationsmodeller.

Artikel A etablerer såkaldte superkritiske afvejninger for resolution, cutting planes, monotone kredsløb og Weisfeiler–Leman. Det er især værd at bemærke, at vores resultater for beviser er superkritiske, selv når de måles i forhold til instansstørrelsen, ikke kun i forhold til antallet af variabler i instansen.

I Artikel B undersøger vi en kompakt kodning af klikformlerne, der udtrykker den modstridende påstand, at en given graf uden en klik af størrelse k indeholder en k-klik. For tilfældigt udvalgte tætte grafer fastlægger vi eksponentielle nedre grænser for cutting planes og resolution over pariteter af begrænset dybde. Vi viser også, at det kræver polynomisk randomiseret kommunikationskompleksitet at finde en falsificeret klausul i disse formler.

Endelig introducerer vi i Artikel C et problem, hvor etableringen af nedre grænser kan ses som et mellemtrin mellem nedre grænser for tilfældige $\Theta(\log n)$-CNF-formler og nedre grænser for pseudorandomgenerator-tautologier af Nisan–Wigderson-typen i bevissystemet cutting planes. Vi beviser eksponentielle nedre grænser for størrelsen af rektangel-DAG'er, der løser dette problem. Til dette formål udvikler vi en ny teknik baseret på approksimationsmetoden.

Acknowledgements

First of all, I want to thank my advisor Jakob Nordström, for his guidance and support, for introducing me to proof complexity, in particular for sharing his passion for supercritical trade-offs and many other fascinating topics within the field, for his thorough feedback on writing and presentations, and for making me a part of the thriving complexity community in Copenhagen.

I thank Robert Robere for hosting me in Montréal during the spring of 2026, and am equally grateful to Dmitry Sokolov for the close collaboration during this time. Thanks to both of you for many interesting discussions at Café Myriade on Saint-Denis.

I am also indebted to Shuo Pang for teaching me many things, from proof complexity to philosophy. I am grateful to Mika Göös for inviting me to spend a stimulating week in Lausanne. Thank you to my Master's thesis advisors Carola Doerr and Johannes Lengler for introducing me to research. I thank my coauthors Anastasia, Artur, David E., Dmitry, Jakob, Noah, Robert, Shuo, Susanna, and Yassine for the fruitful collaborations.

Thank you to all my colleagues in the BARC and MIAO groups, for making this environment an inspiring place. A special thanks to the table tennis athletes, Jacob, Lukas, Sampson, and Shuo; to my travel companions, Amik, Christophe, David E., Gaia, Jonas, Lianna, Kilian, Magnus, Morgan, Noel, Rui, Shuo, and Yassine; the Lund (semi-)affiliates Andy, Benjamin, Christophe, David E., Jonas, Kilian, Morgan, Noel, Rui, Shuo, Wietze, and Yassine, for making the many day trips across the bridge enjoyable; to Joel for always having an open ear, and to Anne for taking care of everyone and everything. Thanks also to Tuukka for pointing me to the pivotal reference [Bod88], and to David L. for proof reading my Danish abstract. I wish to thank Jonathan, Julian, and Sami for their friendship in Copenhagen. For their hospitality during my stay in Montréal, I am grateful to my office mates Ari, Ben, Izzy, Riju, and Stefan, and outside the lab to Christie, Claire, and Maxime.

Finally, I am most grateful to my family for all their help, even from far away Bern, and to Dasha, for her unwavering support during this time, for always making me laugh, and for every moment spent together.

Table of Contents

Abstract ii

Resumé iii

Acknowledgements iv

I Thesis 1

1 Introduction 2

2 Background 5
- 2.1 Notation and Preliminaries . . . 5
- 2.2 Complexity Classes . . . 6
- 2.3 Proof Complexity . . . 7
- 2.4 Circuit Complexity . . . 12
- 2.5 Communication Complexity and Lifting . . . 13
- 2.6 The Weisfeiler–Leman Algorithm . . . 16

3 Contributions 18
- 3.1 Truly Supercritical Trade-offs . . . 18
- 3.2 Average-Case Hardness of Binary Clique . . . 23
- 3.3 Communication Lower Bounds Beyond Jukna's Criterion . . . 26
- 3.4 Discussion . . . 28

Bibliography 31

II Included papers 45

A Truly Supercritical Trade-offs 46

B Average-Case Hardness of Binary-Encoded Clique 97

C Communication Lower Bounds Beyond Jukna's Criterion 129

Part I

Thesis

Chapter 1

Introduction

The topics covered in this thesis lie in *computational complexity theory*, that is, the study of efficient computing. Which *computational tasks* admit efficient algorithms? And which tasks are prohibitively hard? In addition to these questions about *hardness*, one may also ask for *trade-offs* between different resources: is there a task which is solved by a low-memory algorithm, and a different, faster algorithm, but no algorithm can be low-memory and fast *simultaneously*?

Consider, as an example, the following task. You are planning a major international sports tournament, and picking which cities should host the games. To keep things fair for the teams traveling between games, you want every host city to be reachable from the others by a short train ride. Is there a suitable set of ten host cities?

In technical jargon, this is the *clique* problem: given a *graph* and an integer k, does the graph contain a clique of size k? That is, does it have k vertices, which are all mutually connected by an edge? In our example, the integer k is ten, and for the graph, each city is a vertex, and two cities are connected by an edge if they are only a short train ride apart. No efficient algorithms are known for this problem, and for a good reason. The problem is one of Karp's classic NP-complete problems [Kar72], so any *polynomial-time* algorithm for it would contradict the widely held belief that $\mathsf{P} \neq \mathbf{NP}$. Formally ruling out such an efficient algorithm by proving an *impossibility theorem* for its existence, however, is a daunting undertaking: it has been named as one of seven Millennium problems [Mil00], a challenge for all of modern mathematics, whose resolution comes with a million dollar prize.

As a stepping stone towards this overarching goal for their field, compu-

tational complexity theorists seek to establish hardness in weaker models of computation. Examples include restricted types of circuits and query models, but also *mathematical proofs*, which are the focus of this thesis. By proofs, we mean *certificates*, e.g., for the validity of a mathematical theorem, which are *efficiently verifiable*. Getting back to our clique example, an affirmative answer to the problem is easy to certify: we can simply check the train connection between every pair of cities. But what if the answer is no? How can we efficiently certify that no suitable set of ten host cities exists? We could encode this claim into a propositional formula and formally prove its validity. The resources required by such proofs are studied in *proof complexity*.

All the (propositional) proofs appearing in this thesis can be viewed as a sequence of *lines*, each of which is either patently true, or derived from preceding ones using an *inference rule*. If we fix the inference rules and what can be written in a line of the proof, we get a *proof system*. This thesis studies two types of shortcomings of restricted proof systems: lower bounds on the *size* of proofs in that system for specific tautologies, and how different resources such as size, *width* or *depth* interact.

Among the most fundamental questions in proof complexity are the following: Does every (tautological) formula have a short proof in some proof system? That is, does it have a proof in few lines, say, polynomially many in its size? Cook and Reckhow [CR79] connected these questions back to the P vs. NP problem. In particular, they proved that a negative answer to this question for every proof system would imply that $\mathsf{P} \neq \mathsf{NP}$. Again, by curtailing the power of the computational model under study, in this case by studying these questions for very restricted proof systems, we can still hope to obtain impossibility results. Indeed, over the past decades, many results of this kind have appeared for a variety of proof systems, see [Tse68, Hak85, Ajt94, BIK+94, BPR97, Kra97, Pud97, Raz98] for prominent examples. This line of work fits into the *Cook–Reckhow program* for separating NP from coNP (and thus P from NP), which suggests proving lower bounds for increasingly stronger proof systems until one hopefully gains some insight useful for the general case.

Another motivation to study restricted proof systems is their tight connection to algorithms used in practice. By formally modelling how such an algorithm reasons, we can extract a proof system which captures the types of reasoning the algorithm is capable of. This means that any impossibility result or trade-off for the proof system also applies to algorithms restricted to this type of reasoning! For two of the proof systems appearing in this thesis, such

a connection is known: *resolution* captures the *conflict-driven clause learning algorithm* in *satisfiability* (SAT) solving [AFT11, PD11], and *cutting planes* corresponds to its namesake method for integer programming [Gom63, Chv73].

Beyond this connection to more practical domains within computer science, proof systems are also tightly intertwined with other subjects of computational complexity theory. We will use the notion of *communication* throughout our analyses, and exploit tight connections to *circuit* models to transfer results we obtain for proofs to circuits. Furthermore, we will build on and extend work done in the context of the *Weisfeiler–Leman algorithm* [WL68] for graph isomorphism testing.

The papers included in this thesis fit into two broad themes: Paper B and Paper C establish lower bounds on proof size, whereas Paper A gives trade-offs between different resources. In particular, Paper B gives lower bounds for a succinct encoding of the clique problem discussed above in cutting planes and *resolution over parities*, as well as for *randomized communication*. Paper C contributes an intermediate result towards showing that there are *pseudorandom generator tautologies* hard to prove for cutting planes. Lastly, Paper A gives qualitatively new *supercritical* trade-offs for resolution, between the complexity measures width and depth, depth and size, as well as width and size in the treelike case. The results are based on an analogous result for the Weisfeiler–Leman algorithm, and we also transfer it to *monotone circuits* and cutting planes.

The rest of this thesis is structured as follows. In Chapter 2, we introduce the necessary background for our contributions, which we summarize in Chapter 3. After that, Part II consists of the full-length versions of the three papers included in this thesis.

Chapter 2

Background

In this chapter, we provide the necessary background to state and explain the main results included in this thesis. We start by setting up some notation and going over basic preliminaries in Section 2.1, then we recall some complexity classes in Section 2.2. Section 2.3 discusses proof complexity, including the specific proof systems and formulas appearing in this thesis. In Section 2.4, we cover circuits, before diving into communication and its connections to the latter two topics in Section 2.5. Finally, Section 2.6 introduces the Weisfeiler–Leman algorithm.

2.1 Notation and Preliminaries

We use the symbol $\sqcup$ to denote a disjoint union of sets. For an integer k, we write $[k] = \{1, \ldots, k\}$. Given a graph $G = (V, E)$, a subset $H \subseteq V$ is said to be a k-clique if $|H| = k$ and all pairs of distinct vertices in H are connected by an edge. We abbreviate *directed acyclic graph* as DAG. The Erdős-Rényi graph distribution $\mathcal{G}(n, p)$ is obtained by including every possible edge independently with probability p in an undirected n-vertex graph.

We work with *Boolean variables*, taking the values *true* (1) or *false* (0). A well-formed logical expression using Boolean variables, parentheses, as well as the connectives AND ($\wedge$), OR ($\vee$) and NOT ($\neg$), is a *Boolean formula*.

Taking either the Boolean variable x or its negation $\overline{x}$ gives a *literal* over x. A disjunction $C = \ell_1 \vee \cdots \vee \ell_k$ of literals over pairwise distinct variables is called a *clause*, and its *width* is the number of literals k. Combining several clauses into a conjunction $F = C_1 \wedge \cdots \wedge C_m$ yields a *CNF formula*. The

width of a CNF formula F is the maximum width among its clauses; if F is of width k, it is also said to be a *k-CNF formula*.

A mapping from the variables to truth values 0 or 1 (a *truth value assignment*) *satisfies* a Boolean formula if the formula evaluates to 1 under the mapping; if the formula evaluates to 0, it is *falsified* by the mapping. In particular, a CNF formula is satisfied by a truth value assignment if all of its clauses are satisfied by the assignment, which in turn happens if at least one literal in every clause is satisfied. A Boolean formula is said to be *satisfiable* if there is a truth value assignment satisfying it. Otherwise, it is *unsatisfiable*. A formula which is satisfied by every truth value assignment to its variables is called a *tautology*.

For a binary string $x \in \{0,1\}^*$, we denote its length as $|x|$. A subset $L \subseteq \{0,1\}^*$ is called a *language*. The language consisting of all (encodings of) satisfiable Boolean formulas is SAT, while UNSAT consists of all unsatisfiable ones. The tautologies make up the language TAUT.

2.2 Complexity Classes

To make our discussion from the introduction more precise and set the stage for introducing proof complexity, we review some of the complexity classes. For a comprehensive treatment of the subject, we refer to the book [AB09].

Instead of being fully formal and introducing the computational model of the *Turing machine* [Tur37], we will use the term *algorithm* and *running time* in a slightly informal way. We call a function $f\colon \{0,1\}^* \to \{0,1\}^*$ *polynomial-time computable* if there is an algorithm A computing f such that the running time of A on input $x \in \{0,1\}^*$ is polynomial in the length of x. For a language $L \subseteq \{0,1\}^*$, we will consider *decision problems*, which, for a given string $x \in \{0,1\}^*$, ask whether $x \in L$.

The class P contains all languages that have a decision algorithm running in time polynomial in the length of the input string. The class NP consists of the languages such that a yes-solution to their decision problem can be *verified* in polynomial time with a polynomial-size certificate. More formally, $L \subseteq \{0,1\}^*$ is in NP if there is a polynomial-time algorithm A taking as input two strings $x, u \in \{0,1\}^*$, together with a polynomial f, such that $x \in L$ if and only if there is some $u \in \{0,1\}^*$ of length $|u| \leq f(|x|)$ on which $A(x,u) = 1$. The languages in the class coNP are those whose complement lies in NP, i.e., $\mathsf{coNP} = \{L \subseteq \{0,1\}^* \mid \{0,1\}^* \setminus L \in \mathsf{NP}\}$.

It is clear that $\mathsf{P} \subseteq \mathsf{NP}$ and $\mathsf{P} \subseteq \mathsf{coNP}$. The P vs. NP problem asks whether $\mathsf{P} = \mathsf{NP}$. Asking whether the class NP is closed under taking complements, i.e., whether $\mathsf{NP} = \mathsf{coNP}$, is the NP vs. coNP problem. Moreover, $\mathsf{NP} \neq \mathsf{coNP}$ would imply $\mathsf{P} \neq \mathsf{NP}$, since P is closed under taking complements.

We say that a language L *(Karp) reduces* to another language L' if there is a polynomial-time algorithm A with the property that, for all $x \in \{0,1\}^*$, $A(x) \in L'$ if and only if $x \in L$. A language is called *complete* for a class if it is in the class and all other languages in the class reduce to it. The language SAT consists of all (encodings of) satisfiable propositional formulas. Cook [Coo71] and Levin [Lev73] independently proved that SAT is NP-complete. The languages TAUT and UNSAT are coNP-complete.

2.3 Proof Complexity

We first recall the abstract definition of proof systems and their connection to the NP vs. coNP question. Then we review the specific proof systems and propositional formulas appearing in this thesis. A more thorough overview of these topics can be found in [Kra19, BN21].

In their foundational paper, Cook and Reckhow [CR79] gave the following definition of a proof system, upon which the area of proof complexity was built.

Definition 2.3.1 (Proof system [CR79]). A *proof system* for TAUT is a polynomial-time computable function $P\colon \{0,1\}^* \to \{0,1\}^*$ with $\mathrm{im}(P) = \mathsf{TAUT}$.

We call $\pi \in \{0,1\}^*$ with $P(\pi) = F$ a *P-proof* of F.

Let us emphasize three properties of the definition above. The fact that $\mathrm{im}(P) \subseteq \mathsf{TAUT}$ means that P is *sound*, i.e., it can only prove tautologies. On the other hand, $\mathrm{im}(P) \supseteq \mathsf{TAUT}$ ensures that P is *complete*, meaning that P can prove all tautologies. Finally, P is *efficient*, as it is required to be polynomial-time computable.

Cook and Reckhow also introduced the notion of *p-boundedness*. A proof system P is said to be *p-bounded* if there is a polynomial f such that, for every $F \in \mathsf{TAUT}$, there is a P-proof π of F with length $|\pi| \leq f(|F|)$. Determining whether a p-bounded proof system exists is equivalent to solving the NP vs. coNP problem.

Theorem 2.3.2 (Cook–Reckhow [CR79])**.** *There is a p-bounded proof system if and only if* $\mathsf{NP} = \mathsf{coNP}$.

The next notion, *p-simulation*, again due to [CR79], lets us compare the power of proof systems. A proof system P *p-simulates* another proof system Q if there is a polynomial-time computable function $f\colon \{0,1\}^* \to \{0,1\}^*$ such that $P(f(\pi)) = Q(\pi)$ for all $\pi \in \{0,1\}^*$. In particular, if P p-simulates Q, then every family of tautologies with polynomial-length Q-proofs also has polynomial-length P-proofs. This allowed Cook and Reckhow to recast their theorem as a program for proving $\mathsf{NP} \neq \mathsf{coNP}$: for increasingly stronger proof systems P, prove that some family of tautologies does not have polynomial-length P-proofs. Then, the hope is that one could eventually handle the general case.

Finally, let us remark that the proof systems appearing in this thesis, as in much of the proof complexity literature, are *refutation systems*, that is, they are proof systems for UNSAT rather than TAUT.

2.3.1 Proof Systems

Let us now define the concrete proof systems we will encounter over the course of this thesis.

Resolution. The *resolution* proof system [Bla37] operates directly with clauses. Given a CNF formula F, a *resolution refutation* $\pi : F \vdash \bot$ of F is a sequence of clauses $\pi = (C_1, \dots C_\ell)$, whose last clause is the empty clause $C_\ell = \bot$. Furthermore, we require every clause in the sequence to be either an axiom of F, or derived from two preceding clauses using the *resolution rule*:

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

Sometimes it is convenient to also allow the *weakening rule*, which allows one to derive from a clause C any clause $D \supseteq C$. For resolution refutations, weakening steps can be removed at no cost in terms of any complexity measure on the refutation, so this distinction is immaterial.

With every refutation π, we associate a DAG G_π in the following way. For every $i \in [\ell]$, there is a node v_i, and if C_i was derived from C_j, C_k via a resolution step, there are directed edges $(v_j, v_i), (v_k, v_i) \in E(G_\pi)$. Forgetting about the directions of the edges, if G_π is a tree, then the refutation π is said to be *treelike*.

We care about the following complexity measures of resolution refutations. The *size* (sometimes also called *length*) of a resolution refutation π is measured as the length ℓ of the sequence of clauses. The *width* of π is the maximum width among all the clauses in π, i.e., the largest number of literals among the clauses in π. At step i, the *clause space* is defined as one plus the number of clauses C_j, $j < i$, with edges to C_k, $k \geq i$. The additional unit accounts for C_i. The clause space of a resolution refutation π is the largest clause space at any step i, where $1 \leq i \leq \ell$. Finally, the *depth* of π is the length (counted as the number of edges) of the longest path in G_π. We also define all of the measures above for refuting a CNF formula F by minimizing the measure over all resolution refutations π of F. The *trivial* refutation refutes any unsatisfiable CNF formula on n variables in width n, clause space n, depth n, and size $2^{n+1} - 1$ *simultaneously*.

Cutting Planes. The next proof system we consider, *cutting planes* [CCT87], has linear inequalities for its lines. A *cutting planes refutation* of a system of linear inequalities $Ax \geq b$, for $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$ and $x \in \mathbb{Z}^n$, is a sequence of linear inequalities $(d_1 x \geq c_1, \ldots, d_\ell x \geq c_\ell)$, $d_i \in \mathbb{Z}^n$ and $c_i \in \mathbb{Z}$, such that the final inequality is the trivially false $0 \geq 1$, and any inequality occurring in the sequence is either from the system $Ax \geq b$, or derived from preceding ones by one of the following rules:

1. *addition*,
$$\frac{d_i x \geq c_i \qquad d_j x \geq c_j}{(d_i + d_j)x \geq c_i + c_j};$$
2. *multiplication*,
$$\frac{d_i x \geq c_i}{(c \cdot d_i)x \geq c \cdot c_i};$$
for integers $c \geq 0$, or
3. *division*
$$\frac{d_i x \geq c_i}{(d_i / c)x \geq \lceil c_i / c \rceil},$$
where the division is component-wise, and we assume that the integer $c > 0$ divides all entries of d_i.

We also consider a stronger, *semantic* version of cutting planes. Here, the only inference rule is *semantic deduction*, which allows one to derive

any semantically implied inequality from two preceding ones. Specifically, this means that one can derive $d_i x \geq c_i$ from $d_j x \geq c_j$ and $d_k x \geq c_k$, if for any $x \in \{0,1\}^n$ satisfying both of the latter inequalities, x also satisfies the former.

Furthermore, we consider a restriction of cutting planes with *bounded coefficients* [BPR97], i.e., there is a fixed (polynomial in n) bound on every coefficient occurring in any inequality in the proof.

We associate a DAG G_π with a cutting planes refutation π in the same way as for resolution, which again allows us to consider a *treelike* version of the system. The complexity measures *size* (or *length*) and *depth* are defined in the same way as for resolution, both for refutations and for refuting a given instance.

We will only consider cutting planes refutations of CNF formulas, which is enabled by the following translation. We include the Boolean constraints $0 \leq x$ and $x \leq 1$ for every propositional variable x, and translate clauses $C = \bigvee_{i\in I} x_i \vee \bigvee_{j\in J} \overline{x_j}$, into inequalities $\sum_{i\in I} x_i + \sum_{j\in J}(1-x_j) \geq 1$. Cutting planes p-simulates resolution. It can also implement the same *trivial* refutation, and the same upper bounds on depth and size hold for any CNF formula F. The same holds for bounded-coefficient cutting planes. The converse simulation does not hold, since even treelike cutting planes has short refutations of the pigeonhole principle [Juk12].

Resolution over Parities. The final proof system we consider is essentially resolution, but operating with *linear* clauses. That is, the lines of the *resolution over parities* [IS20] proof system are disjunctions of $\mathbb{F}_2$-linear equations. A *linear form* is given as $f_i(x) = \sum_{j=1}^{n} d_{ji} x_j$, with $d_{ji} \in \mathbb{F}_2$. Given *linear equations* $f_i(x) = c_i$, where the variables take values in $\mathbb{F}_2$ and all $c_i \in \mathbb{F}_2$, a *linear clause* is an expression of the form $C = \bigvee_{i=1}^{m}(f_i = c_i)$. We can translate an (ordinary) clause $C = \bigvee_{i\in I} x_i \vee \bigvee_{j\in J} \overline{x_j}$ into the linear clause $C = \bigvee_{i\in I}(x_i = 1) \vee \bigvee_{j\in J}(x_j = 0)$. In this way, we can consider refutations of CNF formulas. A *resolution over parities refutation* of a CNF formula F is a sequence of linear clauses $\pi = (C_1, \ldots, C_\ell)$, such that the final one is the empty clause $C_\ell = \bot$, and each clause is either an axiom of F, or derived from preceding ones by one of the following rules:

1. The *resolution rule*

$$\frac{C \vee (f = 1) \qquad D \vee (f = 0)}{C \vee D},$$

2. or the *semantic weakening rule*, which states that from the linear clause C, one can derive any other linear clause that is semantically implied.

The *size* of a refutation is the number of clauses ℓ in the sequence. We also associate a DAG G_π with a refutation in the same way as before, which allows us to define the *depth* of π analogously.

The main motivation to study this system is that it is the weakest subsystem of $\mathsf{AC}^0[2]$-Frege for which no family of hard CNF formulas is known yet! Recently, there has been a lot of progress on proving lower bounds for restricted subsystems of it [EGI24, AI25, EI25, BBCI26], culminating in an exponential lower bound for refutations of nearly quadratic depth.

Resolution over parities p-simulates resolution; however, some of its bounded-depth subsystems do not [IK26]. The relative power of resolution over parities and cutting planes is an open question.

2.3.2 Propositional Formulas

In this section, we introduce the propositional formulas used throughout this thesis.

Tseitin Formulas. The *Tseitin formulas* were used by Tseitin [Tse68] to prove the first proof complexity lower bounds, and have been a focal point of the field ever since, see [FP22] for a survey. They serve as the basis of our construction in Paper A.

It is based on the *handshaking lemma* from graph theory, and encodes the contradictory claim that the sum of all vertex degrees in a graph is odd. More formally, given a graph $G = (V, E)$, we introduce a variable x_e for every $e \in E$, and assign a *charge* $\chi(v) \in \{0, 1\}$ to each vertex $v \in V$, such that $\sum_{v \in V} \chi(v) = 1 \pmod 2$. Then the Tseitin formula based on G and χ is the CNF formula encoding all the constraints $\sum_{e \ni v} x_e = \chi(v) \pmod 2$ for all $v \in V$. We denote it $\mathrm{Ts}(G)$ if χ is evident from the context or the particular choice is not relevant.

Clique Formulas. The *clique formulas* encode, for a given graph G on n vertices containing no k-clique, that G *has* a k-clique. In Paper B, we study a succinct encoding of this claim, named the *binary clique formulas*.

For a vertex $v \in V(G)$, we denote its representation in binary as $\mathrm{bin}(v) = v_1, \ldots, v_{\log n}$, tacitly assuming that n is a power of two. Let us introduce

propositional variables $x_{i,a}$ for all $1 \leq i \leq k$ and $1 \leq a \leq \log n$, that is, we have $\log n$ variables pointing out a vertex in V for every clique member $i \in [k]$. We write $(x \neq 0)$ for the literal x and $(x \neq 1)$ for the literal $\overline{x}$. The clause

$$\bigvee_{a=1}^{\log n} (x_{i,a} \neq v_a)$$

is denoted $(\vec{x_i} \neq \text{bin}(v))$ and encodes that v is not the ith clique member. Then the formula $\mathsf{BClique}(G, k)$ under study contains the clauses

$$(\vec{x_i} \neq \text{bin}(u)) \vee (\vec{x_j} \neq \text{bin}(v)), \quad \forall i \neq j \in [k], u \neq v \in V : \{u, v\} \notin E \tag{2.1}$$

$$(\vec{x_i} \neq \text{bin}(v)) \vee (\vec{x_j} \neq \text{bin}(v)), \quad \forall i \neq j \in [k], v \in V. \tag{2.2}$$

The first type of clauses (2.1), the so-called *edge axioms*, ensure that no non-edge is in the clique, whereas the second type of clauses (2.2), known as the *functionality axioms*, encode that all vertices in the clique are distinct.

Random Formulas. The third class of CNF formulas that we define, and will consider in Paper C, consists of *random* instances. Formally, we define the distribution $\mathcal{F}(m, n, \Delta)$ as uniformly at random sampling m clauses of width Δ over n variables, with repetitions. The resulting CNF formula is unsatisfiable with high probability if $m \geq \ln 2 \cdot 2^{\Delta} n$ for all $\Delta \geq 3$ [CS88].

2.4 Circuit Complexity

In this section, we recall the definitions of the circuit models we will encounter in this thesis. For an introduction to the area of *circuit complexity*, we direct the reader to Jukna's book [Juk12].

A *monotone Boolean circuit* is a DAG whose nodes (called *gates*) are either labelled by input variables or the functions AND ($\wedge$) or OR ($\vee$). It computes a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ by evaluating the gates in topological order, where the inputs of the functions at a gate are given by its predecessors, and the output value is determined by a distinguished *output gate*. In a (2-ary) *monotone real circuit* [Pud97], the gates may compute any monotone function $f_g\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ of their inputs, whereas input and output values are still in $\{0,1\}$.

The *size* of a circuit is the number of gates it has. Its *depth* is the length of the longest path in the underlying DAG. Note that every monotone Boolean

function is computable via a monotone CNF formula. Therefore, even if we restrict the fan-in to two, every monotone function may be computed in depth $\mathrm{O}(n)$.

2.5 Communication Complexity and Lifting

Communication complexity has close ties to both proof and circuit complexity, which we make ample use of throughout the papers included in this thesis. We only state the definitions we rely on, and sketch the connections to the aforementioned areas. For more details, see the book [RY20] for an overview of communication complexity, and the survey [dRGR22] for its connections to proof and circuit complexity.

Introduced by Yao [Yao79], the *deterministic two-party communication* model involves two players (*Alice* and *Bob*), who are given separate inputs $x \in X$ and $y \in Y$. They have to compute the function $f\colon X \times Y \to \{0,1\}$, while exchanging as few bits as possible. To achieve this, they agree on a *deterministic communication protocol*, which is given by a rooted binary tree Π. At the internal nodes of Π, it is determined who speaks. The spoken bit (which is a function of the node and the input of the player who speaks) determines which child of the node is the next node. The communication then follows this tree, each player speaking when it is their turn, until reaching a leaf. The leaves are labelled by output values. A *randomized communication protocol* is simply a distribution over deterministic protocols, such that, on every input, with constant probability at least $2/3$, the output is correct. We measure the *cost* of deterministic protocols as the depth of the protocol tree; for randomized protocols, we take the maximum cost over the (deterministic) protocols in the support of the distribution.

2.5.1 Search Problems

To connect communication models to proof and circuit complexity, we will use *(total) search problems*. Abstractly, a search problem is a relation $\mathcal{S} \subseteq \mathcal{I} \times \mathcal{O}$. It is said to be *total* if for every $x \in \mathcal{I}$, there is some $o \in \mathcal{O}$ such that $(x, o) \in \mathcal{S}$. The corresponding computational task that we will study is, given $x \in \mathcal{I}$ as an input, output an $o \in \mathcal{O}$ with $(x, o) \in \mathcal{S}$. For all $o \in \mathcal{O}$, we write $\mathcal{S}^{-1}(o) = \{x \in \mathcal{I} \mid (x, o) \in S\}$ for the set of inputs x for which the output o is valid.

We are interested in the following two search problems. For the connection to proof complexity, given an unsatisfiable CNF formula F on n variables, we study the *falsified clause search problem* $\mathsf{Search}(F)$. It is defined to be the relation $\mathsf{Search}(F) \subseteq \{0,1\}^n \times F$ containing all (x, C) where the assignment x falsifies the clause C. For the connection to circuit complexity, we study the *monotone Karchmer–Wigderson game* [KW90]. Given a monotone function f on n variables, it is defined to be the relation $\mathsf{mKW}(f) \subseteq f^{-1}(1) \times f^{-1}(0) \times [n]$, where $\mathsf{mKW}(f)$ contains exactly the (x, y, i) with $x_i > y_i$. Clearly, both of these search problems are total.

To study these problems in a communication setting, we will either write $\mathcal{I}$ as a product set $\mathcal{I} = X \times Y$, or compose them with a *gadget* $g\colon X \times Y \to \mathcal{I}$, as follows: For $\mathcal{S} \subseteq \mathcal{I}^n \times \mathcal{O}$, the composition with g is defined as $\mathcal{S} \circ g^n \subseteq (X^n \times Y^n) \times \mathcal{O}$, where $(x_1, \ldots, x_n, y_1, \ldots, y_n; o) \in \mathcal{S} \circ g^n$ if and only if $(g(x_1, y_1), \ldots, g(x_n, y_n); o) \in \mathcal{S}$.

2.5.2 DAG-like Communication and Query Models

We next introduce the computational models used for solving search problems, and discuss how they relate to the proof and circuit models introduced earlier. We give the abstract definition of *shape-DAGs*, which is due to [GGKS20], and generalizes notions from the earlier works [Raz95, Kra98, Pud10, Sok17, HP18].

Definition 2.5.1 (Shape-DAGs [GGKS20]). For $\mathcal{F}$ a set of functions from $\mathcal{I}$ to $\{0,1\}$, an *$\mathcal{F}$-DAG* solving $\mathcal{S} \subseteq \mathcal{I} \times \mathcal{O}$ is a rooted DAG D of fan-out at most 2, where each node $v \in D$ is labeled by some $f_v \in \mathcal{F}$ such that

1. the root $r \in D$ is labeled by $f_r = 1$ (the constant 1 function);
2. for a node v with children u, w, $f_v^{-1}(1) \subseteq f_u^{-1}(1) \cup f_w^{-1}(1)$; and
3. every leaf ℓ is labelled by an output $o \in \mathcal{O}$ such that $f_\ell^{-1}(1) \subseteq \mathcal{S}^{-1}(o)$.

We call the elements of $\mathcal{F}$ the *shapes* of D. The *size* of D is the number of nodes in D, and its *depth* is the longest root-to-leaf path in D.

Given a bipartite input domain $X \times Y$, we will be particularly interested in the following two shapes. A *(combinatorial) rectangle* is a set of the form $R = R_X \times R_Y$, where $R_X \subseteq X$ and $R_Y \subseteq Y$. A *rectangle-DAG* is an $\mathcal{F}$-DAG, where $\mathcal{F}$ is the set of indicator functions of rectangles in $X \times Y$. A subset

$T \subseteq X \times Y$ is a *triangle* if there are functions $a_T \colon X \to \mathbb{R}$ and $b_T \colon Y \to \mathbb{R}$ such that $T = \{(x, y) \in X \times Y \mid a_T(x) \leq b_T(y)\}$. In particular, every rectangle is a triangle. A *triangle-DAG* is an $\mathcal{F}$-DAG, where $\mathcal{F}$ is the set of indicator functions of triangles in $X \times Y$.

The following proposition allows us to transfer lower bounds from rectangle-DAGs or triangle-DAGs to some of the proof systems and circuit models discussed above.

Proposition 2.5.2 ([Sok17, HP18])**.** *A monotone Boolean circuit computing f yields a rectangle-DAG solving $\mathsf{mKW}(f)$ of the same size and depth.*

A monotone real circuit computing f yields a triangle-DAG solving $\mathsf{mKW}(f)$ of the same size and depth.

A cutting planes refutation of a CNF formula F yields a triangle-DAG solving $\mathsf{Search}(F)$ (for any partition of the variables as $\{0,1\}^n = X \times Y$) of the same size and depth.

We also remark that cutting planes refutations with bounded coefficients can be turned into rectangle-DAGs with at most a polynomial blow-up in size [GGKS20].

Finally, we discuss a *query* model of similar flavour, which gives lower bounds for resolution over parities. Recall that here we are working with the domain $\mathbb{F}_2$. Taking the shapes to be the indicator functions of affine subspaces of $\mathbb{F}_2^n$ yields *affine-subspace-DAGs*. We further restrict this definition with the following two requirements for technical convenience. First, we require every non-leaf node $v \in D$ to have out-degree 2, and that the two out-edges be labelled with $f_v = 0$ and $f_v = 1$. Second, for every edge (v, w), the linear system corresponding to the affine subspace labelling w is implied by the same linear system for v together with the linear equation labelling the edge. If both of the requirements are given, we call D an *affine-DAG*. Lower bounds for affine-DAGs transfer to resolution over parities refutations.

Proposition 2.5.3 ([EGI24])**.** *A resolution over parities refutation of a CNF formula F yields an affine-DAG solving $\mathsf{Search}(F)$ of at most the same size and depth.*

2.5.3 Lifting Theorems

A lifting theorem uses composition with a gadget to *fortify* a problem, whereafter a bound in a weak computational model can be transferred to a bound

for the composed problem in a strong computational model. Many lifting theorems using a variety of gadgets and lifting hardness between all kinds of models have appeared in the literature. We discuss a lifting theorem that features prominently in Paper A. It transfers hardness from resolution width to monotone circuit size.

Theorem 2.5.4 (Width-to-monotone-size lifting (informal) [GGKS20])**.** *Consider a CNF formula F on n variables, and let w denote the minimum resolution width required to refute F. There is a gadget g, whose size is polynomial in n, such that the minimum size of a rectangle-DAG solving $\mathsf{Search}(F) \circ g^n$ is at least $n^{\Omega(w)}$.*

The statement in [GGKS20] is more general; we have specified it to the case most relevant in this thesis. To extract a lower bound on monotone circuit size, we need to reduce $\mathsf{Search}(F) \circ g^n$ to a monotone Karchmer–Wigderson game for some function f. This is possible, since we can extract a rectangle cover of the input domain from $\mathsf{Search}(F) \circ g^n$, which then induces an instance of $\mathsf{mKW}(f)$ for some monotone function f [Raz90, Gál01]. We refer to Robere's thesis [Rob18] for a proof. The same statement also holds for triangle-DAGs, which in turn allows us to lift bounds on the resolution width to bounds on monotone real circuit size for the composed problem, as discussed in the previous section. With some extra work to reduce between *lifted* CNF formulas and gadget-composed falsified clause search problems, we can also derive lower bounds on the size of cutting planes refutations (for the lifted CNF formulas).

2.6 The Weisfeiler–Leman Algorithm

The last topic that we cover is the Weisfeiler–Leman algorithm [WL68] for classifying graphs. It is closely related to proof complexity; see, e.g., [BG15, BN23], and several other areas. In fact, it appears as a subroutine in Babai's monumental quasipolynomial-time graph isomorphism algorithm [Bab16], and finds applications in machine learning [MRF+19, MLM+23]. We refer to [Kie20] for a thorough introduction to this topic.

In its most simple form, known as *color refinement*, the algorithm works as follows. The vertices of a given graph G are initially colored according to their degree. Subsequently, this coloring is refined in iterations, where

vertices are distinguished if the multisets of colors among their neighbors differ. Eventually, no further refinements are made, and the process *stabilizes*.

To obtain the *k-dimensional* Weisfeiler–Leman algorithm, for $k \geq 2$, we look at colorings of k-tuples of vertices in G. (Color refinement is 1-dimensional Weisfeiler–Leman.) We start with a graph $G = (V, E, c)$ together with a vertex coloring $c\colon V \to \mathbb{N}$. As before, the algorithm will iteratively refine a coloring of the k-tuples of vertices, denoted $\chi^{(i)}$ after the ith round. Initially, the k-tuple $\vec{u} = (u_1, \ldots, u_k)$ is colored by its own isomorphism class, i.e., all isomorphic k-tuples get the same color. Note that an isomorphism needs to preserve both edges and the coloring c.

To define the coloring after the ith iteration, we need the following piece of notation. Substituting the jth entry of a k-tuple $\vec{u}$ with $v \in V$ is written as $\vec{u}\ [v/u_j] = (u_1, \ldots, u_{j-1}, v, u_{j+1}, \ldots, u_k)$. The coloring $\chi^{(i)}(\vec{u})$ of $\vec{u}$ after round i is then defined by appending to $\chi^{(i-1)}(\vec{u})$ a *multiset* of tuples:

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

We say that this process *stabilizes* after round t if any two tuples which were colored the same in round t, have the same color again in round $t+1$, i.e., if it holds $\chi^{(t)}(\vec{u}) = \chi^{(t)}(\vec{v})$, then also $\chi^{(t+1)}(\vec{u}) = \chi^{(t+1)}(\vec{v})$. The *iteration number* on G is the minimum t such that the algorithm stabilizes after round t.

If we want to distinguish two graphs G and H using the algorithm, we can simply compare the colorings $\chi^{(i)}(G)$ and $\chi^{(i)}(H)$. It is said that G and H are *distinguished* by k-dimensional Weisfeiler–Leman in t rounds if for any color c, the number of tuples colored by c differs in $\chi^{(t)}(G)$ and $\chi^{(t)}(H)$.

Chapter 3

Contributions

In this chapter, we summarize the contributions of the three papers included in this thesis. The first inclusion, Paper A, presents new or improved trade-offs in proof complexity, circuit complexity, and for the Weisfeiler–Leman algorithm. In Paper B, we study a concise encoding of the k-clique problem, and prove lower bounds for proof and communication complexity. Lastly, in Paper C, we introduce a new technique to prove lower bounds for rectangle-DAGs and use it to prove lower bounds on a problem related to random CNF formulas and pseudorandom generator tautologies.

3.1 Truly Supercritical Trade-offs

This paper studies so-called *supercritical* trade-offs. It establishes the first such trade-off for monotone circuits, as well as a range of new trade-offs in proof complexity and for the Weisfeiler–Leman algorithm. This is the full version of the conference paper [dRFJ+25], which appeared at the *57th Annual ACM Symposium on Theory of Computing (STOC '25)*.

3.1.1 Background

To explain what a supercritical trade-off is, we first describe what we mean by an 'ordinary' trade-off. Let us suppose that we are trading the complexity measures μ and ν, that is, to optimise μ, we have to pay in terms of ν, and vice-versa. Concretely, a trade-off result ideally[1] consists of the following:

[1] If a result consists of all three, it is sometimes called a *true* trade-off in the literature.

1. a worst-case upper bound μ_{worst} on μ;
2. a worst-case upper bound ν_{worst} on ν; and
3. a conditional lower bound on ν that applies if μ is optimized.

Usually, Item 1 and Item 2 are given by some 'trivial' or 'brute-force' procedure. In the 'ordinary' setting, which is depicted in Figure 3.1a, the lower bound in Item 3 stays well below the worst-case upper bound. (In the figure, this is shown by the blue point being to the left of the leftmost horizontal dashed line.)

In his 2016 paper, Razborov [Raz16] proposed a stronger[2] trade-off in proof complexity, termed *supercritical trade-off*. In this setting, the trade-off lower bound on measure ν in Item 3 exceeds the worst-case bound ν_{worst}. (This corresponds to the hatched box extending beyond the dashed horizontal line in Figure 3.1b.) The trade-off Razborov exhibited was quite extreme: There are CNF formulas where optimizing refutation width leads to *doubly-exponential* treelike size, contrasting with the exponential upper bound given by the trivial refutation. Predating Razborov's result, the trade-offs in [BBI16, Ber12, BNT13] are also supercritical, and in the following decade, more such results have appeared in proof complexity [Raz17, Raz18, BN20, FPR22, BT26, CD24] and finite model theory [BN23, GLN25, GLNS25].

One shortcoming shared among many of the results mentioned above (in fact all, except [BBI16, Ber12, BNT13, GLNS25]) is that the trade-off is only supercritical when the worst case upper bound on ν is measured in terms of the number of variables of the instance F. However, the size $|F|$ of these instances is massive, so that if we measure ν_{worst} in terms of $|F|$, the trade-off lower bound on ν (also as measured in terms of $|F|$) does not exceed $\nu_{\text{worst}}(|F|)$.

3.1.2 Main results

In Paper A, we address this issue, proving several supercritical trade-offs where the lower bound in Item 3 is also supercritical when measured in terms of the *instance size* (cf. Figure 3.1c). We call such a trade-off *truly* supercritical.

[2]Indeed, the results were so strong that this phenomenon was called *ultimate* trade-off in an early version of the paper [Raz15b].

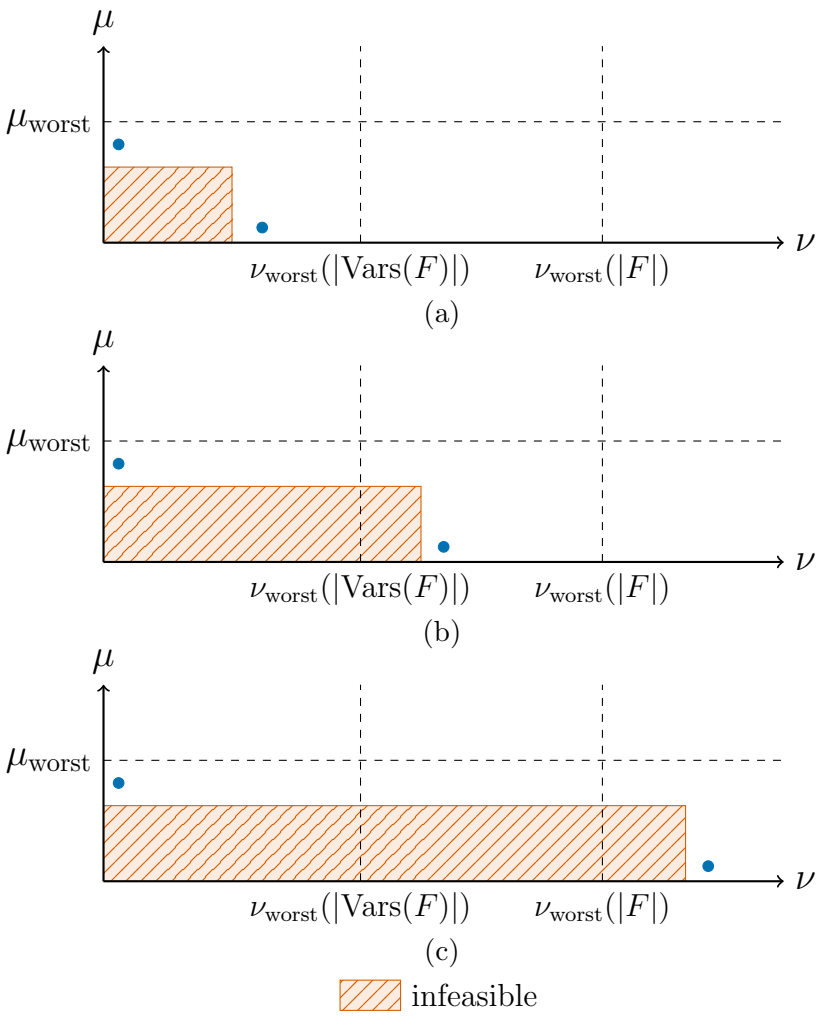


Figure 3.1: A depiction of the different types of trade-offs, adapted from [dRFJ+25]. We show the worst case upper bounds on μ and ν as dashed lines. For ν, we distinguish two worst case upper bounds: first, as measured in terms of the number of variables in the instance, and second, as measured in terms of the instance size. The (blue) dots indicate provable upper bounds for the measures μ and ν. No proof with a combination of measures in the hatched region is possible. Figure 3.1a shows an ordinary (non-supercritical) trade-off, Figure 3.1b shows a supercritical trade-off, and Figure 3.1c shows a truly supercritical trade-off.

We start with a truly supercritical trade-off for resolution width versus depth. This is the first such result where even approximating the optimal width leads to supercritical depth.

Theorem 3.1.1 (Width-depth trade-offs (informal) [dRFJ+25])**.** *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) $N = \text{poly}(n)$ 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 superlinear in $S(F_N)$.

From this result, we derive a truly supercritical trade-off between width and size in treelike resolution.[3]

Theorem 3.1.2 (Width-size trade-offs (informal) [dRFJ+25])**.** *There are CNF formulas F_N of size $S(F_N) = \text{poly}(N)$ over N variables such that:*

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

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

We also obtain trade-offs between size and depth, for both resolution and cutting planes proofs.

Theorem 3.1.3 (Cutting planes trade-offs (informal) [dRFJ+25])**.** *There are 3-CNF formulas F_N of size $S(F_N)$ over N variables such that:*

(1) Resolution refutes F_N in size $S(F_N \vdash \bot)$ polynomial in N, but any cutting planes refutation with size at most $S(F_N \vdash \bot)^{1.4}$ must have depth at least $S(F_N)^{2.4}$.

[3] A similar result was obtained in [BLV25] prior to our work for treelike resolution, but building on our Theorem 3.1.1.

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

Furthermore, the fact that our trade-off in Theorem 3.1.1 is truly supercritical makes an improved lifting theorem based on Theorem 2.5.4 applicable, which leads to the first supercritical trade-offs for monotone circuits.

Theorem 3.1.4 (Monotone circuit trade-offs (informal) [dRFJ+25])**.** *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 of size at most $s \cdot \exp\left((\log N)^{1.9}\right)$ computing f_N must have depth superpolynomial in N.

We also mention concurrent work [GMRS25], which obtains similar trade-offs to the results mentioned above.

Finally, we get a trade-off for Weisfeiler–Leman dimension versus iteration number.

Theorem 3.1.5 (Weisfeiler–Leman trade-offs (informal) [dRFJ+25])**.** *For all c, k with $1 \leq c \leq k-1$ 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.*

This trade-off is actually the technical starting point of this work, as we explain next.

3.1.3 Techniques

Grohe, Lichter, Neuen, and Schweitzer [GLNS25] introduced *compressed CFI graphs* to prove a lower bound on the iteration number of the k-dimensional Weisfeiler–Leman algorithm (which can be viewed as a trade-off between dimension and iteration number). Running the Weisfeiler–Leman algorithm on these graphs is analyzed using a new *compressed* version of the classical

treewidth cop-robber game [ST93], which was already used in the previous best-known lower bound due to Fürer [Für01]. It is known that for the (uncompressed) cop-robber game played on a graph G, every play that ends in a cop win corresponds to a resolution refutation π of the Tseitin formula $\mathrm{Ts}(G)$, where the number of cops in play corresponds to the width of π, and the number of rounds until the cops win corresponds to the depth of π [GTT20]. By appropriately defining *compressed* Tseitin formulas, we get a similar correspondence for the compressed game. However, due to a slight mismatch (for Weisfeiler–Leman, the cops are placed on *vertices*, whereas for resolution, the cops stay on the *edges* of G), we need the robust trade-off in Theorem 3.1.5 to derive a trade-off for resolution. Our result is based on a refinement of the graph compression in [GLNS25] and its analysis.

To derive our other trade-off results, we use (new or improved) lifting theorems, as introduced in Section 2.5. For Theorem 3.1.2, we prove a lifting theorem using the XOR gadget. A crucial point is that the width increases after lifting. To derive our size-depth trade-offs in Theorem 3.1.3 and Theorem 3.1.4, we would like to apply a lifting in the style of Theorem 2.5.4. However, since the trade-off lower bound in our width-depth trade-off only applies to a small range of widths, we need to be careful about the constant in the exponent of the lower bound given by Theorem 2.5.4. This is why the previous work [GGKS20, LMM+22] does not apply directly. Instead, we prove a lifting theorem where the constant in the exponent is nearly tight, whose proof is based on a new way to cover a combinatorial triangle by *structured* rectangles. We additionally provide a tighter lifting theorem for resolution, based on a simple random restriction argument.

3.2 Average-Case Hardness of Binary Clique

In Paper B, we study the binary encoding of the clique formula. We show average-case hardness for cutting planes, bounded-depth resolution over parities, as well as randomized communication. This is the full version of a conference paper [dREG+26] that appeared at the *53rd EATCS International Colloquium on Automata, Languages, and Programming (ICALP '26)*.

3.2.1 Background

The k-clique problem, as discussed in the introduction, is widely believed to be exceedingly hard to solve: not only is it one of the 21 original NP-complete problems [Kar72], even just approximating an optimal solution up to an $n^{1-\varepsilon}$-factor is NP-hard [Hås99, Zuc07]. However, rigorously proving that the k-clique problem admits no efficient algorithms is prohibitively ambitious, as it would resolve the P vs. NP problem.

Instead, complexity theorists are looking for evidence of this hardness, such as at least establishing it for more restricted models of computing. Indeed, some of the first lower bounds for monotone circuits [Raz85, AB87] were proven for the k-clique problem, and later extended in [CKR21, BM25, dRV25]. Rossman [Ros08, Ros14] showed *average-case* lower bounds on the k-clique problem for both monotone and bounded-depth circuits.

In Paper B, we focus on restricted models within proof complexity. More precisely, let us suppose that a graph G does not contain a clique of size k. Now, we encode the (contradictory) claim that G *does* contain a clique of size k as a CNF formula, and study the refutations of this formula. Here, particular attention needs to be paid to how the k-clique problem is encoded as a CNF formula. Both *unary* and *binary* encodings are studied in the literature, where the binary case is usually harder to refute for proof systems, and thus easier to prove lower bounds against. Previous work concerning the unary encoding shows strong average-case lower bounds for regular resolution [ABdR$^+$21] and *unary Sherali–Adams* [dRPR23]. Since the publication of our paper, this has also been shown for treelike cutting planes [dREGR26]. For the more challenging binary encoding, similar bounds are known even for (general) resolution [LPRT17] and ℓ-DNF resolution [DGGM24], where $\ell = o(\log \log n)$.

3.2.2 Main results

The main contribution of Paper B are lower bounds for the binary encoding of the k-clique problem in the cutting planes and bounded-depth resolution over parities proof systems. We show exponential size lower bounds in the average case. The results are stated separately, as the hidden parameters differ.

Theorem 3.2.1 (Informal, [dREG$^+$26])**.** *For graphs G on n vertices sampled according to an Erdős-Rényi distribution with appropriate edge density, any cutting planes refutation of* $\mathsf{BClique}(G, k)$ *must have size at least* $2^{n^{\Omega(1)}}$.

Theorem 3.2.2 (Informal, [dREG+26]). *For graphs G on n vertices sampled according to an Erdős-Rényi distribution with appropriate edge density, any resolution over parities refutation of* $\mathsf{BClique}(G, k)$ *in depth $m^{1.5-\varepsilon}$, m being the number of variables in* $\mathsf{BClique}(G, k)$*, must have size at least $2^{n^{\Omega(1)}}$.*

We supplement these results with an analogous lower bound in communication complexity, for the falsified clause search problem over $\mathsf{BClique}(G, k)$.

Theorem 3.2.3 (Informal, [dREG+26]). *For graphs G on n vertices sampled according to an Erdős-Rényi distribution with appropriate edge density, the randomized communication complexity of* $\mathrm{Search}(\mathsf{BClique}(G, k))$ *is $n^{\Omega(1)}$.*

Note that the polynomial lower bound on communication complexity is indeed analogous to an exponential lower bound on proof size, as the trivial protocol, where either party just sends their entire input to the other, has linear communication cost. Theorem 3.2.3 also implies an exponential size lower bound for treelike cutting planes on $\mathsf{BClique}(G, k)$, which is subsumed by Theorem 3.2.1.

3.2.3 Techniques

All of our results have fairly independent proofs, so we will discuss the techniques for each of them separately. A common theme among all the proofs in the paper is that they are based on techniques for analogous lower bounds on the *binary pigeonhole principles* (BPHP).

For our cutting planes result, we apply the framework of bottleneck counting. It originates in circuit complexity [HC99], and was adapted for cutting planes by Sokolov [Sok24]. Later it was also applied to BPHP [BW25]. We start by extracting a triangle-DAG Π from the cutting planes refutation using Proposition 2.5.2. Then we construct a partial function $\mu\colon X \times Y \to V(\Pi)$, mapping inputs to nodes in Π. If we can show that μ has large domain, and that every node of Π has a small preimage under μ, we can conclude that Π must have many nodes. As a technical contribution, we identify a combinatorial property of random graphs, which suffices to carry out the argument.

For bounded-depth resolution over parities, we apply the *random-walks-with-restarts* framework introduced in [AI25], and applied to BPHP in [BI25]. First, we extract an affine-DAG from the resolution over parities refutation. Then we start our (top-down) random walk by sampling a random assignment.

We show that, with noticeable probability, this walk can continue for a significant number of steps without reaching a falsified clause. At its endpoint, we apply the *closure of a set of linear forms* [EGI24] to restrict only a few variables in a way that restores the invariant which guarantees that the walk can be restarted. Assuming that the size of the affine-DAG is small, we show that this process can be repeated many times, hence implying a lower bound on the depth of Π. In our setting, preserving the invariant requires each newly assigned vertex to lie in the common neighbourhood of the vertices already assigned.

Finally, to obtain our lower bounds on randomized communication, by Yao's principle, it suffices to bound the distributional *deterministic* communication cost. We then apply a structural theorem of [GGJL25], which streamlines the previous work of [YZ24] on BPHP, and allows us to assume that the protocol is very *structured*. This allows us to finish the proof by analyzing two cases, distinguishing whether or not the protocol fixes the location of one end of the missing edge within the graph, separately.

3.3 Communication Lower Bounds Beyond Jukna's Criterion

This paper introduces a problem for which proving cutting planes lower bounds can be viewed as an intermediate step between proving cutting planes lower bounds for random $\Theta(\log n)$-CNF formulas and proving similar bounds for *pseudorandom generator tautologies*. We give a lower bound on the rectangle-DAG size for this problem, using a new technique based on the approximation method. This is a manuscript in progress [JRSS26].

3.3.1 Background

While proof complexity has been successful in proving lower bounds for weak proof systems, the same task is considered out of reach for strong proof systems like Frege or Extended Frege. One reason for this is the lack of candidate formulas that could be hard for these proof systems; consider as an example the Tseitin formulas: a strong enough proof system will be able to do parity reasoning and will therefore have short refutations of them. Random CNF formulas have been put forth as a potential candidate and extensively

studied since, see [CS88, BP96, BI99, ABE02, SBI04, Ale11, HP17, FPPR22, Sok24, SS25].

A related approach constructs tautologies based on *pseudorandom generators* [ABRW04, Kra01]; see also [Kra25]. A mapping $G_n\colon \{0,1\}^n \to \{0,1\}^m$, for $m > n$, is said to be a pseudorandom generator *hard for a proof system* P if P cannot efficiently certify that $b \notin \mathrm{Im}(G_n)$ for *every* $b \in \{0,1\}^m$. We will be particularly interested in constructions of such mappings G in the style of the Nisan–Wigderson generator. Concretely, fix an underlying bipartite *dependency graph* $B = (L, R, E)$ such that $|L| = m$ and $|R| = n$, L and R are ordered, and the degree of every vertex in L is Δ. We identify every vertex $i \in R$ with an input variable x_i, and write down a function $f_j\colon \{0,1\}^\Delta \to \{0,1\}$ for every vertex $j \in L$. The mapping G_n is then defined as

$$G_n(x) = (f_1(x \restriction_{N_B(1)}), \ldots, f_m(x \restriction_{N_B(m)})).$$

Razborov [Raz15a] conjectured that some instances of these generators are hard for Frege and Extended Frege. It is known that, for particular encodings and parameter ranges, these generators are hard for resolution and polynomial calculus [ABRW04], as well as k-DNF resolution [Raz15a, Sok22]. For cutting planes, no lower bounds are known.

3.3.2 Technical Contributions

To prove that some Nisan–Wigderson-style generators are hard for cutting planes, we first need to find suitable functions f_i to use in the construction. Alekhnovich et al. [ABRW04] choose the f_i's to be parities of their input bits. This yields tautologies which are essentially unsatisfiable systems of linear equations modulo two. Unfortunately, this will not work for cutting planes, which has quasipolynomial-size refutations of any such system [DT20]. A natural next candidate for the functions f_i would be the *inner product* function $\mathsf{IP}\colon \{0,1\}^{\Delta/2} \times \{0,1\}^{\Delta/2} \to \{0,1\}$, defined as $\mathsf{IP}(x, y) = \sum_i x_i y_i$.

However, all currently known techniques for proving lower bounds on the cutting planes proof system, including *feasible interpolation* [BPR97, Pud97] and its adaptations for random $\Theta(\log n)$-CNF formulas [HP17, FPPR22], *lifting* [GGKS20, LMM+22], as well as *bottleneck counting* [Sok24] seem to be inherently limited for this task; we defer a detailed discussion to the introduction in the manuscript.

Let us now describe the problem that we consider in [JRSS26]. In analogy

to the distribution of random CNF formulas $\mathcal{F}(m, n, k)$, we write $\mathcal{P}(m, n, k)$ for the distribution sampling m random parity constraints of width k over n variables. Then, we let $(\mathcal{F} \otimes \mathcal{P})(m, n, k)$ be the distribution where we first sample $F^X = \bigwedge_i C_i \sim \mathcal{F}(m, n, k)$ and $G^Y = \bigwedge_i P_i \sim \mathcal{P}(m, n, k)$ over disjoint variable sets X and Y, and then output $\varphi = \bigwedge_i (C_i(x) \vee P_i(y))$. Our main result is a lower bound on the size of rectangle-DAGs searching for a falsified constraint in $\varphi \sim (\mathcal{F} \otimes \mathcal{P})(m, n, \log n)$.

Theorem 3.3.1 ([JRSS26])**.** *Let $\alpha, d > 0$, be constants and $\Delta = d \log n$, $m = \alpha n 2^{\Delta}$. For $\varphi \sim (\mathcal{F} \otimes \mathcal{P})(m, n, \Delta)$, a rectangle-DAG Π solving $\mathsf{Search}(\varphi)$ has size at least $2^{\Omega(n)}$ with high probability.*

This implies the same lower bound for refuting φ in bounded-coefficient cutting planes.

Note that the falsified constraint search problem for $\varphi \sim (\mathcal{F} \otimes \mathcal{P})(m, n, k)$ is easier than the falsified clause search problem for $F \sim \mathcal{F}(m, n, k)$, since parity constraints have more falsifying assignments than clauses. Therefore, we view proving lower bounds as in Theorem 3.3.1 as an intermediate step between lower bounds for random $\Theta(\log n)$-CNF formulas and lower bounds for Nisan–Wigderson-style generators based on inner product functions. Indeed, the known techniques to prove lower bounds for cutting planes refutations of random $\Theta(\log n)$-CNF formulas based on Jukna's [Juk97] monotone lower bound criterion and bottleneck counting [HC99, Sok24] already fail for this task. To overcome this, we adopt Razborov's method of approximations directly to rectangle-DAGs [Raz85]. We are currently working on extending our proof to also work for triangle-DAGs.

3.4 Discussion

We have seen trade-offs and lower bounds for proof systems and communication models useful for studying them, monotone circuits, as well as the Weisfeiler–Leman algorithm. We conclude the treatment of our contributions with two potential directions for further work below.

More on supercritical trade-offs. As previously mentioned, the concurrent work [GMRS25] also obtains supercritical trade-offs, but for a different parameter regime. In particular, their result constructs CNF formulas refutable in width $w_{\min} = O(\log n)$, but such that a supercritical depth lower

bound applies to all refutations within width n^ε, for some constant $\varepsilon > 0$. It would be interesting to see a construction unifying their parameter range with ours, getting closer to the generous range of parameters to which the previous supercritical trade-offs [Raz16, FPR22, BT26] (which were supercritical only when measured in terms of number of variables) apply. Concretely, we ask for a truly supercritical trade-off between width and depth that holds for refutations of width $w_{\min} = O(1)$ all the way up to n^ε, for some constant $\varepsilon > 0$. Notably, such a result cannot be obtained purely from a variable compression applied to a Tseitin formula: By combining the results of [Bod88, GTT20] we have that if $\mathrm{Ts}(G)$ has a resolution refutation in width k, then it also has a resolution refutation in width $3k$ and depth $O(\log|\mathrm{Vars}(\mathrm{Ts}(G))|)$. In particular, no *robust* trade-off can hold for a compressed version of $\mathrm{Ts}(G)$.

Another interesting technical challenge arises for supercritical trade-offs in resolution over parities. Since Paper A was published, supercritical trade-offs appeared for resolution over parities size versus depth [IK26, IPS26]. While these results are not truly supercritical, the lifting techniques developed in these papers imply that such a result could be lifted from a strong enough width-depth trade-off in resolution. In particular, this would be implied by a family of formulas of size linear in the number of variables n, refutable by resolution in size $\mathrm{poly}(n)$, but such that any refutation in width $w_{\min} = \Omega(n^{0.51})$ must have truly supercritical depth.

Finally, we would like to see a *truly* supercritical trade-off between width and clause space in resolution.[4] That is, exhibit a family of CNF formulas F_n over n variables, for all $n \in \mathbb{N}$, refutable in small width $w_{\min}$, but such that any resolution refutation in width $w_{\min}$ must have superlinear clause space $\omega(|F_n|)$. This would *qualitatively* generalize our result, since the clause space for a resolution proof is always a lower bound on its depth. Another motivation for such a result comes from the lifting techniques developed in our paper [dRFJ+25], which could be applied to lift a resolution width-space trade-off to a *size*-space trade-off in resolution and cutting planes.

Techniques for cutting planes lower bounds. The next problem we discuss goes back at least as far as the reference [BP98], and has been reiterated many times since. Namely, we would like to see new techniques for proving

[4] A trade-off between these measures that is supercritical when measured in terms of the number of variables was proved in [BN20], based on Razborov's hardness condensation technique [Raz16].

cutting planes lower bounds. Since the first lower bounds for cutting planes were proved via monotone feasible interpolation [BPR97, Kra97, Pud97], some developments have occurred: feasible interpolation was generalized to work for non-split formulas [HP17, FPPR22]; there are now lifting theorems from resolution width to cutting planes size [GGKS20]; and Sokolov adapted bottleneck counting to triangle-DAGs [Sok24]. If our result from Paper C can be extended to triangle-DAGs, this would arguably be a further addition.

However, all of the techniques mentioned above fall short for the following well-known problems for cutting planes: proving lower bounds for random O(1)-CNF formulas (cf. [HP17]), and proving a size (or, more precisely, length) separation between cutting planes with bounded coefficients and unrestricted cutting planes.[5] This seems to be because all known techniques argue about communication complexity, implicitly or explicitly. Therefore, it would be interesting to see a technique which does not have this feature.

[5] See [dRMN+20] for the only known result in this direction.

Bibliography

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

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press Cambridge, April 2009.

[ABdR+21] Albert Atserias, Ilario Bonacina, Susanna F. de Rezende, Massimo Lauria, Jakob Nordström, and Alexander Razborov. Clique is hard on average for regular resolution. *Journal of the ACM*, 68(4):23:1–23:26, August 2021. Preliminary version in *STOC '18*.

[ABE02] Albert Atserias, Maria Luisa Bonet, and Juan Luis Esteban. Lower bounds for the weak pigeonhole principle and random formulas beyond resolution. *Information and Computation*, 176(2):136–152, 2002.

[ABRW04] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Pseudorandom generators in propositional proof complexity. *SIAM Journal on Computing*, 34(1):67–88, 2004. Preliminary version in *FOCS '00*.

[AFT11] Albert Atserias, Johannes Klaus Fichte, and Marc Thurley. Clause-learning algorithms with many restarts and bounded-width resolution. *Journal of Artificial Intelligence Research*, 40:353–373, January 2011. Preliminary version in *SAT '09*.

[AI25] Yaroslav Alekseev and Dmitry Itsykson. Lifting to bounded-depth and regular resolutions over parities via games. In *Pro-*

ceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC '25), pages 584–595, June 2025.

[Ajt94] Miklós Ajtai. The complexity of the pigeonhole principle. *Combinatorica*, 14(4):417–433, 1994. Preliminary version in *FOCS '88*.

[Ale11] Michael Alekhnovich. Lower bounds for k-DNF resolution on random 3-CNFs. *Computational Complexity*, 20(4):597–614, December 2011. Preliminary version in *STOC '05*.

[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.

[BBCI26] Sreejata Kishor Bhattacharya, Farzan Byramji, Arkadev Chattopadhyay, and Russell Impagliazzo. Lower bounds for near-quadratic-depth resolution over parities. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing (STOC '26)*, pages 955–966, June 2026.

[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.

[BG15] Christoph Berkholz and Martin Grohe. Limitations of algebraic approaches to graph isomorphism testing. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP '15)*, volume 9134 of *Lecture Notes in Computer Science*, pages 155–166. Springer, July 2015.

[BI99] Eli Ben-Sasson and Russell Impagliazzo. Random CNF's are hard for the polynomial calculus. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science (FOCS '99)*, pages 415–421, October 1999.

[BI25] Farzan Byramji and Russell Impagliazzo. Lower bounds for bit pigeonhole principles in bounded-depth resolution over parities. *arXiv preprint arXiv:2511.20023*, November 2025.

[BIK+94] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS '94)*, pages 794–806, November 1994.

[Bla37] Archie Blake. *Canonical Expressions in Boolean Algebra*. PhD thesis, University of Chicago, 1937.

[BLV25] Christoph Berkholz, Moritz Lichter, and Harry Vinall-Smeeth. Supercritical size-width tree-like resolution trade-offs for graph isomorphism. In *Proceedings of the 50th International Symposium on Mathematical Foundations of Computer Science (MFCS '25)*, volume 345 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 18:1–18:19, August 2025.

[BM25] Jarosław Błasiok and Linus Meierhöfer. Hardness of Clique Approximation for Monotone Circuits. In *40th Computational Complexity Conference (CCC 2025)*, volume 339 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:20, July 2025.

[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.

[Bod88] Hans L. Bodlaender. NC-algorithms for graphs with small treewidth. In J. van Leeuwen, editor, *Proceedings of the 14th International Workshop on Graph-Theoretic Concepts in Computer Science (WG '88)*, pages 1–10, June 1988.

[BP96] Paul Beame and Toniann Pitassi. Simplified and improved resolution lower bounds. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS '96)*, pages 274–282, October 1996.

[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.

[BPR97] María Bonet, Toniann Pitassi, and Ran Raz. Lower bounds for cutting planes proofs with small coefficients. *Journal of Symbolic Logic*, 62(3):708–728, September 1997. Preliminary version in *STOC '95*.

[BT26] Sam Buss and Neil Thapen. A simple supercritical tradeoff between size and height in resolution. *Information Processing Letters*, 191:106589, January 2026.

[BW25] Paul Beame and Michael Whitmeyer. Multiparty communication complexity of collision-finding and cutting planes proofs of concise pigeonhole principles. In *Proceedings of the 52nd International Colloquium on Automata, Languages, and Programming (ICALP '25)*, volume 334 of *LIPIcs*, pages 21:1–21:20, 2025.

[CCT87] William Cook, Collette Rene Coullard, and György Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, November 1987.

[CD24] Arkadev Chattopadhyay and Pavel Dvořák. Super-critical tradeoffs in resolution over parities via lifting. In *Proceedings of the*

40th Computational Complexity Conference (CCC '25), volume 339 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:19, August 2024.

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

[CKR21] Bruno Pasqualotto Cavalar, Mrinal Kumar, and Benjamin Rossman. Monotone circuit lower bounds from robust sunflowers. In *Proceedings of the 14th Latin American Symposium on Theoretical Informatics (LATIN '20)*, volume 12118 of *Lecture Notes in Computer Science*, pages 311–322. Springer, January 2021.

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC '71)*, pages 151–158, May 1971.

[CR79] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, March 1979. Preliminary version in *STOC '74*.

[CS88] Vašek Chvátal and Endre Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, October 1988.

[DGGM24] Stefan Dantchev, Nicola Galesi, Abdul Ghani, and Barnaby Martin. Proof complexity and the binary encoding of combinatorial principles. *SIAM Journal on Computing*, 53(3):764–802, 2024.

[dREG+26] Susanna F. de Rezende, David Engström, Yassine Ghannane, Duri Andrea Janett, and Artur Riazanov. Average-Case Hardness of Binary-Encoded Clique in Proof and Communication Complexity. In *53rd International Colloquium on Automata, Languages, and Programming (ICALP '26)*, volume 374 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 151:1–151:23, July 2026.

[dREGR26] Susanna F. de Rezende, David Engström, Yassine Ghannane, and Kilian Risse. Superpolynomial length lower bounds for tree-like semantic proof systems with bounded line size. In *Proceedings of the 67th IEEE Symposium on Foundations of Computer Science (FOCS '26)*, 2026. To appear.

[dRFJ+25] Susanna F. de Rezende, Noah Fleming, Duri Andrea Janett, Jakob Nordström, and Shuo Pang. 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)*, pages 1371–1382, June 2025.

[dRGR22] Susanna F de Rezende, Mika Göös, and Robert Robere. Guest column: Proofs, circuits, and communication. *ACM SIGACT News*, 53(1):59–82, March 2022.

[dRMN+20] Susanna F. de Rezende, Or Meir, Jakob Nordström, Toniann Pitassi, Robert Robere, and Marc Vinyals. Lifting with simple gadgets and applications to circuit and proof complexity. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science (FOCS '20)*, pages 24–30, November 2020.

[dRPR23] Susanna de Rezende, Aaron Potechin, and Kilian Risse. Clique is hard on average for unary Sherali-Adams. In *Proceedings of the 64th IEEE Annual Symposium on Foundations of Computer Science (FOCS '23)*, pages 12–25, 2023.

[dRV25] Susanna F. de Rezende and Marc Vinyals. Lifting with Colourful Sunflowers. In Srikanth Srinivasan, editor, *40th Computational Complexity Conference (CCC '25)*, volume 339 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 36:1–36:19, July 2025.

[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.

[EGI24] Klim Efremenko, Michal Garlík, and Dmitry Itsykson. Lower bounds for regular resolution over parities. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24)*, pages 640–651, 2024.

[EI25] Klim Efremenko and Dmitry Itsykson. Amortized closure and its applications in lifting for resolution over parities. In *Proceedings*

of the 40th Computational Complexity Conference (CCC '25), pages 8:1–8:24, 2025.

[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*.

[GGJL25] Mika Göös, Tom Gur, Siddhartha Jain, and Jiawei Li. Quantum communication advantage in TFNP. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, (STOC '25)*, pages 1465–1475, 2025.

[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*.

[GLN25] Martin Grohe, Moritz Lichter, and Daniel Neuen. The iteration number of the Weisfeiler-Leman algorithm. *ACM Transactions*

on Computational Logic, 26(1):6:1–6:31, January 2025. Preliminary version in *LICS '23*.

[GLNS25] Martin Grohe, Moritz Lichter, Daniel Neuen, and Pascal Schweitzer. Compressing CFI graphs and lower bounds for the Weisfeiler-Leman refinements. *Journal of the ACM*, 72(3):1–27, June 2025. Preliminary version in *FOCS '23*.

[GMRS25] Mika Göös, Gilbert Maystre, Kilian Risse, and Dmitry Sokolov. Supercritical tradeoffs for monotone circuits. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC '25)*, pages 1359–1370, June 2025.

[Gom63] Ralph E. Gomory. An algorithm for integer solutions of linear programs. In R.L. Graves and P. Wolfe, editors, *Recent Advances in Mathematical Programming*, pages 269–302. McGraw-Hill, New York, 1963.

[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*.

[Hak85] Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39(2-3):297–308, August 1985.

[Hås99] Johan Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182:105–142, 1999. Preliminary version in *FOCS '96*.

[HC99] Armin Haken and Stephen A. Cook. An exponential lower bound for the size of monotone real circuits. *Journal of Computer and System Sciences*, 58(2):326–335, April 1999.

[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.

[HP18] Pavel Hrubeš and Pavel Pudlák. A note on monotone real circuits. *Information Processing Letters*, 131:15–19, March 2018.

[IK26] Dmitry Itsykson and Alexander Knop. Supercritical Tradeoff Between Size and Depth for Resolution over Parities. In Shubhangi Saraf, editor, *17th Innovations in Theoretical Computer Science Conference (ITCS '26)*, volume 362 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 81:1–81:20, Dagstuhl, Germany, January 2026.

[IPS26] Dmitry Itsykson, Vladimir Podolskii, and Alexander Shekhovtsov. Resolution Width Lifts to Near-Quadratic-Depth Res($\oplus$) Size. In Dana Moshkovitz, editor, *41st Computational Complexity Conference (CCC '26)*, volume 383 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 13:1–13:27, July 2026.

[IS20] Dmitry Itsykson and Dmitry Sokolov. Resolution over linear equations modulo two. *Annals of Pure and Applied Logic*, 171(1):102722, 2020. Preliminary version in *MFCS '14*.

[JRSS26] Duri Andrea Janett, Robert Robere, Anastasia Sofronova, and Dmitry Sokolov. Communication lower bounds beyond jukna's criterion. Manuscript, August 2026.

[Juk97] Stasys Jukna. Finite limits and monotone computations: The lower bounds criterion. In *Proceedings of the 12th Annual IEEE Conference on Computational Complexity (CCC '97)*, pages 302–313. IEEE Computer Society, June 1997.

[Juk12] Stasys Jukna. *Boolean function complexity: advances and frontiers*, volume 27 of *Algorithms and Combinatorics*. Springer, January 2012.

[Kar72] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Springer, 1972.

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

[Kra97] Jan Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *Journal of Symbolic Logic*, 62(2):457–486, June 1997.

[Kra98] Jan Krajíček. Interpolation by a game. *Mathematical Logic Quarterly*, 44(4):450–458, 1998.

[Kra01] Jan Krajíček. On the weak pigeonhole principle. *Fundamenta Mathematicae*, 170(1-3):123–140, 2001.

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

[Kra25] Jan Krajíček. *Proof Complexity Generators*. London Mathematical Society Lecture Note Series. Cambridge University Press, Cambridge, May 2025.

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

[Lev73] Leonid A. Levin. Universal sequential search problems. *Problemy peredachi informatsii*, 9(3):115–116, 1973. In Russian. Available at http://mi.mathnet.ru/ppi914.

[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.

[LPRT17] Massimo Lauria, Pavel Pudlák, Vojtěch Rödl, and Neil Thapen. The complexity of proving that a graph is Ramsey. *Combinatorica*, 37(2):253–268, April 2017. Preliminary version in *ICALP '13*.

[Mil00] The Millennium Problems of the Clay Mathematics Institute, May 2000. See https://www.claymath.org/millennium-problems.

[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.

[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.

[PD11] Knot Pipatsrisawat and Adnan Darwiche. On the power of clause-learning SAT solvers as resolution engines. *Artificial Intelligence*, 175(2):512–525, February 2011. Preliminary version in *CP '09*.

[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.

[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.

[Raz98] Alexander A. Razborov. Lower bounds for the polynomial calculus. *Computational Complexity*, 7(4):291–324, December 1998.

[Raz15a] Alexander A. Razborov. Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. *Annals of Mathematics*, 181(2):415–472, March 2015.

[Raz15b] Alexander A. Razborov. An ultimate trade-off in propositional proof complexity. Technical Report TR15-033, Electronic Colloquium on Computational Complexity (ECCC), March 2015.

[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.

[Rob18] Robert Robere. *Unified Lower Bounds for Monotone Computation*. PhD thesis, University of Toronto, 2018. Available at https://hdl.handle.net/1807/92007.

[Ros08] Benjamin Rossman. On the constant-depth complexity of k-clique. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC '08)*, pages 721–730, 2008.

[Ros14] Benjamin Rossman. The monotone complexity of k-clique on random graphs. *SIAM Journal on Computing*, 43(1):256–279, 2014. Preliminary version in *FOCS '10*.

[RY20] Anup Rao and Amir Yehudayoff. *Communication Complexity and Applications*. Cambridge University Press, January 2020.

[SBI04] Nathan Segerlind, Samuel R. Buss, and Russell Impagliazzo. A switching lemma for small restrictions and lower bounds for k-DNF resolution. *SIAM Journal on Computing*, 33(5):1171–1200, 2004. Preliminary version in *FOCS '02*.

[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.

[Sok22] Dmitry Sokolov. Pseudorandom Generators, Resolution and Heavy Width. In *Proceedings of the 37th Computational Complexity Conference (CCC '22)*, volume 234 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:22, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[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.

[SS25] Anastasia Sofronova and Dmitry Sokolov. A lower bound for k-dnf resolution on random CNF formulas via expansion. In *Proceedings of the 40th Computational Complexity Conference, CCC '25, Toronto, Canada, August 5-8, 2025*, volume 339 of *LIPIcs*, pages 32:1–32:27. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, August 2025.

[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. The complexity of a deduction in the propositional predicate calculus. *Zapiski Nauchnyh Seminarov Leningradskogo Otdelenija matematicheskogo Instituta im. V. A. Steklova akademii Nauk SSSR (LOMI)*, 8:234–259, 1968. In Russian.

[Tur37] Alan M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1937.

[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.

[Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing (STOC '79)*, pages 209–213, New York, NY, USA, April 1979. Association for Computing Machinery.

[YZ24] Guangxu Yang and Jiapeng Zhang. Communication lower bounds for collision problems via density increment arguments. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24)*, pages 630–639, 2024.

[Zuc07] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(6):103–128, August 2007. Preliminary version in *STOC '06*.

Part II

Included papers

Paper A

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

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

Abstract

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

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 [Raz16] 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 [BBI16, Ber12, BNT13, Raz16, Raz17, Raz18, BN20, FPR22, BT26, CD24] (including the works [BBI16, Ber12, BNT13] predating [Raz16]) and finite model theory [BN23, GLN25]. Recent papers [GGKS20, FGI+21, FPR22] 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.

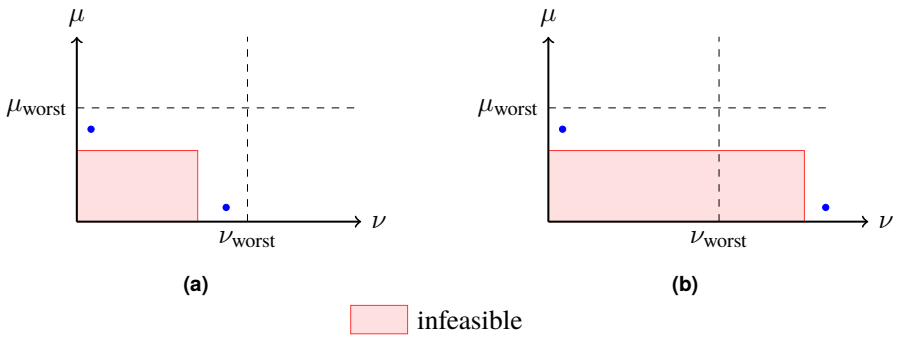


Figure 1: An illustration of trade-offs. Blue dots represent provable upper bounds on measures μ and ν. Proofs with measures in the shaded region are ruled out by the trade-off, where μ_{worst} and ν_{worst} are the worst-case upper bound on μ and ν, respectively. Figure 1a illustrates a non-supercritical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

1.1 Supercritical Trade-offs in Circuit Complexity

In circuit complexity, different measures of the cost associated with computing 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 [Edm65], many questions about this problem remain unresolved, in particular, regarding its monotone complexity. In a breakthrough result in 1985, Razborov [Raz85] 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 [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, Razborov's lower bound remained the state of the art. 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 $2^{\mathrm{O}(n)}$. For a long time, it was not known whether the

lower bound on the size could be improved, or if there were better upper bounds.[1] This led to the following discussion motivating supercritical trade-offs. 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 already know that if there is a monotone circuit of size $n^{\mathrm{O}(\log n)}$ that computes bipartite perfect matching then it must have depth at least $\Omega(n)$. But what if any monotone circuit of size $n^{\mathrm{O}(\log n)}$ 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 [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 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 Supercritical 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 $\mathrm{mod}\ 2$ linear equations—have played a pivotal role. In particular, they were used to prove the first proof complexity lower bounds in [Tse68] (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 [FP22] 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 [CCT87] 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 been established for a variety of formulas [Pud97, BPR97, FPPR22, HP17, Sok24, GGKS20], determining the complexity of cutting planes refutations of Tseitin formulas remained open.

In a surprising turn of events, Dadush and Tiwari [DT20] 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 [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, BT26], 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

[1] After the conference version of this paper appeared, Cavalar, Göös, Riazanov, Sofronova, and Sokolov [CGR+26] proved an exponential lower bound on the monotone circuit complexity of matching.

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. 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 $S(F_N \vdash \bot)$ polynomial in N, but any cutting planes refutation with size at most $S(F_N \vdash \bot)^{1.4}$ must have depth at least $S(F_N)^{2.4}$.

(2) Resolution refutes F_N in size $S(F_N \vdash \bot)$ quasi-polynomial in N, but any cutting planes refutation with size at most $S(F_N \vdash \bot) \cdot \exp\big((\log N)^{1.9}\big)$ 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 [BBI16, Ber12, BNT13], 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 [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 (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 $W(F_N \vdash \bot) = \mathrm{o}(\log N)$, but any treelike refutation with width at most $1.4 \cdot W(F_N \vdash \bot)$ must have size at least $\exp\big(S(F_N)^{2.4}\big)$.

(2) Resolution refutes F_N in width $W(F_N \vdash \bot) = \mathrm{o}\big((\log N)^{3/2}\big)$, but any treelike resolution refutation with width at most $W(F_N \vdash \bot) + 40\frac{\log N}{\log\log N}$ 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:*

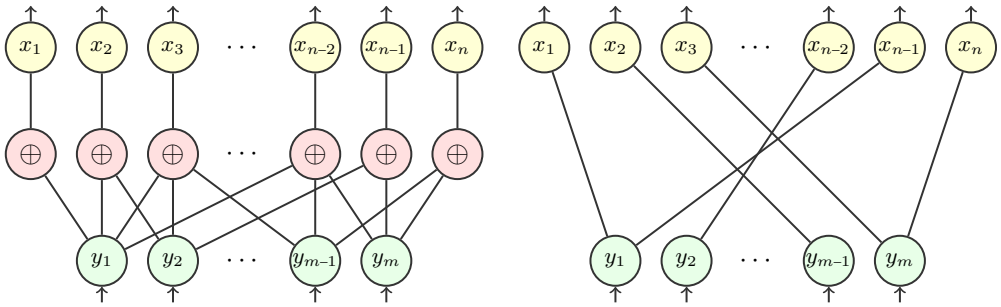
(1) $N = \mathrm{poly}(n)$ and any refutation of width at most $w + C$ has depth exponential in $\mathrm{poly}(S(F_N))$.

(2) $N = \mathrm{o}(2^{n/2})$ 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 [Ber12]. 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 the Weisfeiler–Leman Algorithm

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 [Bab16], is related to machine learning [MRF+19, Gro21, MLM+23], and has also turned out to be relevant for 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



(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 $m \ll n$.

(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, GLN25]. For a long time, the best lower bound was linear [Für01] until the works [BN23, GLN25] 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. 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 $1 \leq c \leq k-1$ 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 [CFI92], 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* [Raz16], 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 [BN23] to prove the Weisfeiler–Leman trade-offs discussed above.

The result of the recent paper [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, the trade-offs in [GLNS23] 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 [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* 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 [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.

1.5 Related Work

In concurrent work, Göös, Maystre, Risse and Sokolov [GMRS25] 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 $\mathrm{O}(\log n)$, 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 $n^{\mathrm{O}(\log n)}$ 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 [GMRS25]. 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 [BLV25] obtained a truly supercritical width-size trade-off for treelike resolution. We showed our treelike resolution trade-off only after the results in [BLV25], 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. 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. For completeness, we provide the translation to the Weisfeiler–Leman algorithm in Appendix A and a proof of the Full Image Lemma in Appendix B.

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. The term *minimal* is always used with respect to the set-inclusion relation.

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* [ST93]. 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.

In Section 3 and Section 4, we will analyze a variant of the game called the *compressed cop-robber game* [GLNS23]. It is played on a graph endowed with equivalence relations on both vertices and edges as follows.

Definition 2.1 (Graph compression). 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 compression leads to the *compressed cop-robber game*. In this game, intuitively, the cops have copies 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 copy. The formal definition is given in Section 3.1.

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 3.2. 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 novelty of our analysis compared to [GLNS23] 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 [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 $\chi^{(i-1)}(\vec{u})$:

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

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 (see Appendix A), Theorem 2.2 gives the following trade-off for Weisfeiler–Leman algorithms, which 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 $\big(2^{-(c+10)}k^{-3}n\big)^{k/(c+1)}$ 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], 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 $1 \leq c \leq k-1$, 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 $\big(2^{-(c+10)}k^{-3}n\big)^{k/(c+1)}$ 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 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. Sometimes we write $z \in \{0,1\}^{|Vars(F)|}$ for a *total assignment* of the variables of F, and denote by $C(z) = 1$ or $C(z) = 0$ that the clause C evaluates to true or false under z, respectively.

The *Tseitin formula* [Tse68] 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. A vertex labeled 1 is said to have an *odd charge*. 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.,

$$\sum_{e \ni v} x_e = \chi(v). \tag{2.1}$$

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*

$$\frac{C \vee x \qquad D \vee \overline{x}}{C \vee D}. \tag{2.2}$$

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 for Resolution

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 $N \in [2k^2 n^{c+1}, 24k^2(2n)^{c+1}]$ variables, having formula size $\mathrm{O}(N)$, which resolution can refute in width $k + 3$ and size $\mathrm{O}(k^2(4n)^k)$ simultaneously, but for which any refutation of width at most $k + c - 1$ 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 a translation between the compressed cop-robber game and resolution refutations of Tseitin formulas. For the uncompressed game, such a connection is known in the literature [GTT20].

The two concrete settings of parameters in Theorem 1.4 are obtained 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 Tight Lifting and Supercritical Trade-offs for Resolution

We obtain our other proof and circuit complexity trade-offs from the width-depth trade-off via *lifting*. Lifting is a framework for composing functions. We refer to the inner functions as *gadgets*. For a CNF formula, 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:

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

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 2.6 (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 of the XOR-substituted formula $\mathrm{XOR}_m(F)$, 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 2.7 (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 $N \in [2k^2 n^{\lfloor \varepsilon k \rfloor} m, 24k^2(2n)^{\lfloor \varepsilon k \rfloor} m]$ variables, having formula size $\mathrm{O}(16^m \cdot N)$, 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 $k+3$, simulating this refutation line-by-line while maintaining all clauses resulting from the substitution $x_i = y_{i,1} \oplus \ldots \oplus y_{i,m}$ 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 $\log s = \Omega(n^k)$. □

The two concrete parameter settings in Theorem 1.3 are obtained 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 [BHP10].[2] 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 then we expand it into a CNF formula. 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 $j \in [m]$. To keep the width of the formula $\mathrm{IND}_m(F)$ small, we instead use extension variables $w_{i,1},\ldots,w_{i,m-1}$ to encode each of these clauses as m clauses of width 3, as follows: we include the clause $x_{i,1} \vee w_{i,1}$, the clauses $\overline{w}_{i,j-1} \vee x_{i,j} \vee w_{i,j}$ for all $1 < j < m$, and the clause $\overline{w}_{i,m-1} \vee x_{i,m}$.

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.

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

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), and the proof, which we defer to Section 6, is simple and based on a random restriction argument.

Simulating a resolution refutation of F step-by-step, we obtain an upper bound on the size required to refute $\mathrm{IND}_m(F)$ in resolution. We include the proof for the sake of completeness.

Lemma 2.9. *For any $m, n \geq 1$ and a CNF formula F over n variables, 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. 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 those axioms in $\mathrm{IND}_m(F)$ containing the extension variables $w_{i,j}$, which can be done in $\mathrm{O}(nm)$ steps.

Our goal is to simulate Π step by step. For every clause $C = \bigvee_{\ell \in [w']} z_{i_\ell}^{\beta_\ell}$ in Π, we keep the invariant that we derive, for each $J = (j_1, \ldots, j_{w'}) \in [m]^{w'}$, the clause $C_J = \bigvee_{\ell \in [w']} (\overline{x}_{i_\ell, j_\ell} \vee y_{i_\ell, j_\ell}^{\beta_\ell})$. If C is an axiom, this holds by definition of $\mathrm{IND}_m(F)$. Suppose the invariant holds for the clauses $C \vee z_i$ 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 the variable $y_{i,j}$. Finally, we can derive $(D \vee C)_J$ in m steps by resolving $(D \vee C)_J \vee \overline{x}_{i,j}$ with $\bigvee_{j \in [m]} x_{i,j}$ for all $j \in [m]$. This gives a total of $m^{w'+1} + m = \mathrm{O}(m^{w+1})$ steps per clause in Π, and hence the claimed size upper bound follows. □

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 $N = \mathrm{O}(mk^2(2n)^{c+1})$ 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 $\mathrm{O}(k^2 (4n)^k)$, 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 bound follows from combining the lifting result in Theorem 2.8 with the width-depth trade-off in Theorem 2.5. □

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

In this section, we discuss our results for monotone circuits and cutting planes proofs. 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, with $x_1 \wedge x_2 = \min\{x_1, x_2\}$ and $x_1 \vee x_2 = \max\{x_1, x_2\}$.

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 *length* 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 for previous DAG-like lifting theorems, it will be convenient to work with the following top-down definitions of these models discussed above—*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 $(x, o) \in \mathcal{S}$. For a given output $o \in \mathcal{O}$, we denote the set of inputs to which o is a solution of $\mathcal{S}$ as $\mathcal{S}^{-1}(o) = \{x \in \mathcal{D} \mid (x, o) \in \mathcal{S}\}$. We start by defining *shape-DAGs* [GGKS20], which are a generalisation of rectangle-DAGs introduced in [Raz95] and simplified in [Pud10, Sok17]. The *fan-out* of a vertex v in a DAG is the number of outgoing edges from v.

Definition 2.11 (Shape-DAG). Let $\mathcal{F} \subseteq 2^{\mathcal{D}}$ 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-out at most 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 *vertex 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

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

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 purpose, 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 $((x, y), i) \in \mathsf{Search}(F) \iff C_i((x, y)) = 0$.

Triangle-DAGs are now related to semantic cutting planes as follows: For any CNF formula F, a semantic cutting planes refutation of F implies that there is, for any partition of the variables of F, a triangle-DAG solving $\mathsf{Search}^{X,Y}(F)$, of the same depth and vertex size equal to its length. Any inequality $az \geq b$ occurring in the refutation can be written as $a_1 x + a_2 y \geq b$. The triangle T defined by $a_T(x) = a_1 x - b$ and $b_T(y) = -a_2 y$ contains exactly the points falsifying the inequality. Associating every in the DAG underlying the cutting planes proof with a triangle in this way and reversing the edges yields a triangle-DAG solving $\mathsf{Search}^{X,Y}(F)$, which satisfies the claims on depth and vertex size by construction. Similarly, a resolution refutation of F implies a rectangle-DAG for $\mathsf{Search}^{X,Y}(F)$ of the same depth and vertex size equal to its length: associate with every clause the rectangle of all points falsifying that clause.

Turning our attention to monotone circuits, we consider the following search problem: 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

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

In words, given two inputs x and y to f, such that $f(x) = 1$ and $f(y) = 0$, the goal is to output a coordinate i such that $x_i = 1$ and $y_i = 0$. The coordinate exists due to the monotonicity of f. The

DAG-like version of the monotone Karchmer–Wigderson relation [Raz95, Kra98, Pud10, Sok17, HP18] implies that there is a monotone circuit (respectively, a monotone real circuit) computing f if and only if there is a rectangle-DAG (respectively, a 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, including the extension variables $w_{i,j}$, and Y to the y-variables of $\mathrm{IND}_m(F)$. It can be reduced to the composed search problem via the following proposition, which has a simple proof independent of the model of communication that can be found in [HN12].

Proposition 2.12. *If there is a rectangle-DAG (respectively, triangle-DAG) solving* $\mathsf{Search}^{X,Y}(\mathrm{IND}_m(F))$*, then there is a rectangle-DAG (respectively, triangle-DAG) solving* $\mathsf{Search}(F) \circ \mathrm{IND}_m^n$ *of the same size and depth.*

Finally, we would like to connect the two search problems discussed above. This is possible via the following reduction [Raz90, Gál01]: First, any covering of a bipartite input domain by rectangles induces a monotone Karchmer–Wigderson $\mathsf{mKW}(f)$ search problem over the same domain for some (partial) monotone function f, as described in, e.g., [Rob18, Section 2.3]. Next, for a composed falsified clause search problem $\mathsf{Search}^{X,Y}(F) \circ g^n$, one can construct a rectangle covering of $X \times Y$ such that the search problem $\mathsf{Search}^{X,Y}(F) \circ g^n$ *is the same as* $\mathsf{mKW}(f)$, up to relabelling, see, e.g., [Rob18, Section 3.2].

The fact below summarizes the translations between search problems we discussed above.

Fact 2.13. *Let F be an unsatisfiable k-CNF formula 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. *An $\mathcal{F}$-DAG solving $\mathsf{mKW}(f)$ implies that there is an $\mathcal{F}$-DAG solving $\mathsf{Search}(F) \circ \mathrm{IND}_m^n$ of the same size and depth.*
2. *An $\mathcal{F}$-DAG solving $\mathsf{Search}(F) \circ \mathrm{IND}_m^n$ implies that there is an $\mathcal{F}$-DAG solving $\mathsf{mKW}(f)$ of the same size and depth.*

The resulting (partial) monotone function f can be extended to an instance of monotone CSP-SAT [RM99, GP18, Oli15], see also [Rob18, Section 6.1].

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

Theorem 2.14 (Lifting for triangle-DAGs). *Let F be an unsatisfiable CNF formula over n variables, and let $m, w \in \mathbb{N}, \delta > 0$ be such that $w \le n$, $0 < \delta < 1 - \frac{1}{\log m}$ and $m \ge (50n/\delta)^{2/\delta}$. 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 in Theorem 2.5, we obtain the supercritical size-depth trade-offs for monotone (real) circuits.

Theorem 2.15 (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 $3 \le c \le k - 1 < \frac{n}{2 \log n}$ and $m \ge (50n/\delta)^{2/\delta}$, 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 Fact 2.13 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 $\mathrm{O}(k^2 (4n)^k)$, 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 that there is a rectangle-DAG solving $\mathsf{Search}^{X,Y}(\mathrm{IND}_m(F))$ of the same size, together with Proposition 2.12 and Item 2 of Fact 2.13. 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 Item 1 of Fact 2.13, there is a triangle-DAG solving $\mathsf{Search}(F) \circ \mathrm{IND}_m^n$ of the same size and depth. Finally, combining the lifting result for triangle-DAGs in Theorem 2.14 and the width-depth trade-off in 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 2.16 (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 $N = \mathrm{O}(m^4k^2(2n)^{c+1})$ 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 2.15, by applying the lifting result in Theorem 2.14 to the width-depth trade-off in Theorem 2.5 together with Fact 2.13, 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 $\widetilde{\mathrm{IND}}_m(F)$. The idea is that the dependence between x and y variables can be captured by 2-clauses, and constraints that are only over x-variables or only over y-variables can be written in 3-CNF by using extension variables.

We first note that by a 3-CNF encoding of a clause $C = \bigvee_{i\in[k]} \ell_i$, where ℓ_i's are literals over only x-variables or only y-variables, we mean any encoding using extension variables that does not have more than $O(k)$ clauses, for example, the following 3-CNF formula

$$(\ell_1 \vee e_{C,1}) \wedge \left(\bigwedge_{i\in[k-1]} (\neg e_{C,i} \vee \ell_{i+1} \vee e_{C,i+1}) \right) \wedge \neg e_{C,k-1} \ell_k \,, \tag{2.7}$$

where $e_{C,i}$ are extension variables. Any extension variable included in the encoding will belong to the same player as the variables of C.

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 variables $x_{C,J}$ and $y_{C,J}$ for every $C \in F$ and every $J \in [m]^{W(C)}$, along with extension variables needed to write clauses as 3-CNF formulas. The variables $x_{i,j}$ encode the pointer variables in unary, that is, $x_{i,j}$ is true if Alice's ith pointer is j. The variable $x_{C,J}$ encodes the pointers to all the variables in C, that is, for $J = (j_1, \ldots, j_{W(C)})$ the variable $x_{C,J}$ will be true if for the first variable of C, Alice's pointer is j_1, for the second variable of C, Alice's pointer is j_2, and so on.

The clauses in $\widetilde{\mathrm{IND}}_m(F)$ consist of

1. a 3-CNF encoding of $\bigvee_{j\in[m]} x_{i,j}$ for every $i \in [n]$, enforcing that Alice must have a pointer for each index, and
2. 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) a 3-CNF encoding of $(\bigwedge_{\ell\in[w]} x_{i_\ell,j_\ell}) \to x_{C,J}$, enforcing that $x_{C,J}$ is true whenever J contains the pointers to the variables in C,
   (b) a 2-clause $x_{C,J} \to y_{C,J}$, and
   (c) a 3-CNF encoding of $y_{C,J} \to \bigvee_{\ell\in[w]} y_{i_\ell,j_\ell}^{\beta_\ell}$, enforcing that if $y_{C,J}$ is true, the variables pointed to by J should satisfy C.

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 record two basic facts about $\widetilde{\mathrm{IND}}_m(F)$. Firstly, upper bounds for $\mathrm{IND}_m(F)$ can be transferred to $\widetilde{\mathrm{IND}}_m(F)$.

Claim 2.17. 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 $\widetilde{\mathrm{IND}}_m(F)$.

Proof. This holds since $\mathrm{IND}_m(F)$ can be derived from $\widetilde{\mathrm{IND}}_m(F)$ in linear size by resolving over the new variables introduced in $\widetilde{\mathrm{IND}}_m(F)$. □

Secondly, there is a reduction between $\mathsf{Search}^{X,Y}(\mathrm{IND}_m(F))$ and $\mathsf{Search}^{\widetilde{X},\widetilde{Y}}(\widetilde{\mathrm{IND}}_m(F))$.

Claim 2.18. 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 $\widetilde{\mathrm{IND}}_m(F)$.

Proof. 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, if $\mathcal{F}$ is a shape-DAG closed under taking intersections with rectangles, then given an $\mathcal{F}$-DAG $\widetilde{\Gamma}$ 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. This implies that Γ is an $\mathcal{F}$-DAG for $\mathsf{Search}^{X,Y}(\mathrm{IND}_m(F))$. □

We are now ready to prove Theorem 2.16.

Proof of Theorem 2.16. 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 $\mathrm{O}(m^4k^2(2n)^{c+1})$ variables and clauses. Since by Theorem 2.5, F has a resolution refutation of width $k+3$ and size $\mathrm{O}(k^2(4n)^k)$, we have by Lemma 2.9 that $\mathrm{IND}_m(F)$, and hence by Claim 2.17 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 $\mathsf{Search}^{\widetilde{X},\widetilde{Y}}(\widetilde{\mathrm{IND}}_m(F))$ of the same size and depth, and by Claim 2.18 there is also a triangle-DAG solving $\mathsf{Search}^{X,Y}(\mathrm{IND}_m(F))$ of size s and depth d. By Proposition 2.12 and Fact 2.13 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 in Theorem 2.14 and the width-depth trade-off in 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 2.15, and Theorem 1.2 from Theorem 2.16. 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 Compressed Cop-Robber Game and Tseitin Formulas

In this section, we discuss the compressed cop-robber game introduced by [GLNS23]. We give its formal definition in Section 3.1. In Section 3.2, we define the specific family of graph compressions we use to establish the lower bound on the number of rounds in Theorem 2.2, and in Section 3.3, we describe the family of formulas exhibiting the width-depth trade-off in Theorem 2.5. We establish the connection between the cop-robber game and the family of formulas in Section 3.4.

3.1 The Compressed Cop-Robber Game

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 [GLNS23]). 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:

(G1) If there is no lifted cop, the cops 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 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 (G1) 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 $w_1 \neq w_2$.)

(G3) 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.

We remark that, as the definition of a compressible move M requires that $M \subseteq E$ is closed under $\equiv_E$, so is $V(M)$ under $\equiv_V$. This implies that condition (G2)b is equivalent to requiring that $\mathbb{C}_{\equiv_V} \cap V(M)_{\equiv_V} = \emptyset$. Consider the trivial case where all vertices form singleton equivalence classes. Then all edges also form singleton equivalence classes, and a compressible move from w_1 to w_2 is a path connecting w_1 and w_2 together with some edge disjoint cycles, which can be removed without violating any of the conditions in the definition.

As mentioned earlier, we analyze CNF formulas associated with a specific family of graph compressions, which we introduce next.

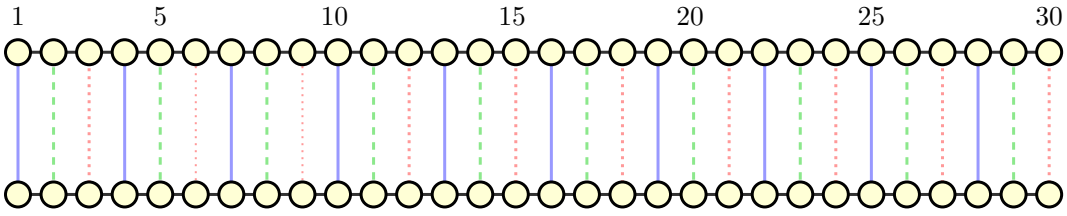
3.2 Compressed Cylinders

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. Let $W_{\equiv_V}$ be the set of all vertices equivalent to some vertex in W, i.e., $W_{\equiv_V} = \{v \mid v \equiv_V w \text{ for some } w \in W\}$. 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 let $F_{\equiv_E}$ be the set of all edges equivalent to some edge in F, i.e., $F_{\equiv_E} = \{e \mid e \equiv_E f \text{ for some } f \in F\}$. For singleton sets of vertices and edges, we write $\{e\}_{\equiv_E}$ as $[e]$ and $\{v\}_{\equiv_V}$ as $[v]$.

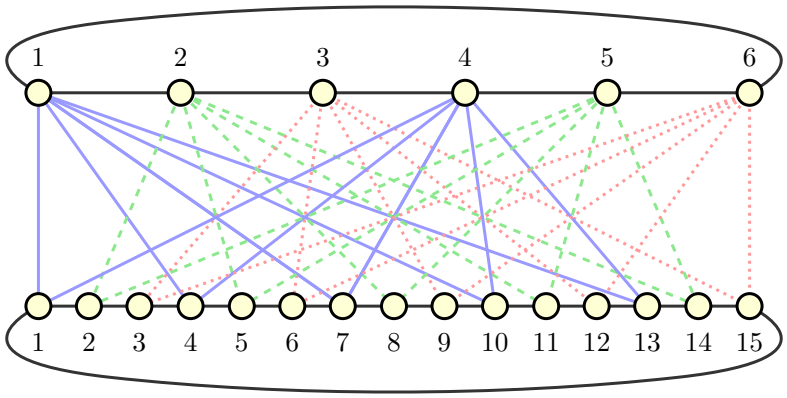
Our construction is based on a compression of the cylinder graph.

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 $(G_{\text{cyl}}, \equiv_V, \equiv_E)$ 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 $E(G_{\text{cyl}})$ 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.



(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.

3. *Choice of parameters.* We set m_i, L, and r in G_{cyl} as follows. Take k pairwise coprime numbers $P_1, \dots, P_k$ 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:

$$m_i = 2(k+c) \cdot P_i \cdots P_{i+c}\,, \tag{3.1}$$

$$L = 2(k+c) \cdot P_1 \cdots P_k\,, \tag{3.2}$$

$$r = k + c + 1\,. \tag{3.3}$$

This definition extends the construction due to Grohe et al. [GLNS23] by letting the row periods depend on the parameter c. Their original construction is recovered in the case when $c = 1$.

Let us make some observations about the compression.

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.

We can visualize the compression as follows. Fixing a row i, we denote the vertex (i, a) as 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 $\{1\}, \dots, \{r\}, \{L + r + 1\}, \dots, \{L + 2r\}$, and $(r+1)_{\equiv_V}, \dots, (r+m_i)_{\equiv_V}$, where $x_{\equiv_V} = \{y \in [r+1, r+L] \mid y \equiv x \mod m_i\}$.

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, and $H_i(1), \dots, H_i(m_i)$, where $H_i(a) = \{\{x, x+1\} \mid x \in [r, r+L],\ x \equiv a \mod m_i\}$. Note that the two edges $\{r, r+1\}$ and $\{r+L, r+L+1\}$ that connect the middle part to the left and right part 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 $i+1$ (where the addition is 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 $g_i = \gcd(m_i, m_{i+1})$ vertical edge classes $T_i(1), \ldots, T_i(g_i)$, where

$$T_i(a) = \left\{ \{(i,x),(i+1,x)\} \;\middle|\; x \in [r+1, r+L] \text{ and } x - a \equiv 0 \mod g_i \right\}. \tag{3.4}$$

See Figure 3 for an illustration.

The number of vertex classes and edge classes satisfy

$$|V(G_{\text{cyl}})/\equiv_V| = 2kr + \sum_{i=1}^{k} m_i; \tag{3.5}$$

$$|E(G_{\text{cyl}})/\equiv_E| = (8r-2)k + \sum_{i=1}^{k} (m_i + \gcd(m_i, m_{i+1})). \tag{3.6}$$

Furthermore, from the condition $P_1, \ldots, P_k \in [n, 2n]$ in Item 3 of Definition 3.2, we get the estimate

$$2kn^{c+1} \leq m_i \leq 4k(2n)^{c+1}. \tag{3.7}$$

We collect some structural properties of the parameters for future reference.

Property 3.3 (Parameters in compression). 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 Item 3 of Definition 3.2.

- (P1) $g_{[k]} \geq 2(k+c)$;
- (P2) For all $i \in [k]$ and all $a, b \geq 1$ such that $a + b \leq c + 1$, $m_i \mid \operatorname{lcm}\left(g_{[i-a,i]},\ g_{[i,i+b]}\right)$. 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 $\operatorname{lcm}(m_i \mid i \in I) = L$;
- (P4) $r > k + c$.

Let us verify that the parameters indeed satisfy the properties above. For (P1) and (P4), it follows immediately from the choice of m_i and r in (3.1) and (3.3). For (P2), again by (3.1) 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 $\operatorname{lcm}(A, B)$ 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 (3.1), so if $|I| \geq k - c$, then P_i is in m_j for some $j \in I$ and hence $\operatorname{lcm}(m_i \mid i \in I) = 2(k+c)P_1 \cdots P_k = L$.

3.3 Compressed Tseitin Formulas

Turning our attention to CNF formulas, we define the *compressed Tseitin formulas*, for which we prove our width-depth trade-off.

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 $x_e \mapsto x_{[e]}$ as follows: for each edge equivalence class, introduce a single, new variable $x_{[e]}$. Replace every occurrence of a variable x_e in $\mathrm{Ts}(G)$ with the variable $x_{[e]}$. We call the resulting CNF formula the *compressed Tseitin formula on* $(G, \equiv_V, \equiv_E)$, and denote it by $\mathrm{Ts}(G)_{\equiv}$.

Remark 3.5. The compressed Tseitin formula has width at most $\deg(G)$, the maximal vertex degree, and it has $|E(G)/\equiv_E|$ variables. Also, the parity constraints at equivalent vertices become the same after the substitution, and so the compressed formula has at most $2^{\deg(G)-1}|V(G)/\equiv_V|$ clauses.

The compressed formula is 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 can be turned into a refutation of the compressed formula by applying the substitution to the refutation. This does not increase the width depth and size of the

refutation. Therefore, the width, depth, and size complexity of the compressed formula is at most that of the original formula.

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 $\text{Ts}(G_{\text{cyl}})_{\equiv}$ as in Definition 3.4. We show our width-depth trade-off in Theorem 2.5 for the formulas obtained from this construction.

We say that a Tseitin formula is *nice with respect to a graph compression* $(G, \equiv_V, \equiv_E)$, if the charges at v, v' are the same if $v \equiv_V v'$, and there is a total assignment $\alpha_{\text{Ts}} : E(G)/_{\equiv_E} \to \mathbb{F}_2$ that satisfies all axioms except for at one vertex class. Note that the instance $\text{Ts}(G_{\text{cyl}})_{\equiv}$ is nice with respect to $(G_{\text{cyl}}, \equiv_V, \equiv_E)$, with the total assignment being all-zero and the unique vertex class with odd charge being $(1, 1)$, which is a singleton class. Together with Remark 3.5, (3.5), (3.6), and (3.7) imply that the compressed Tseitin formula $\text{Ts}(G_{\text{cyl}})_{\equiv}$ has width 4, and consists of N variables and $\Theta(N)$ clauses, for $2k^2 n^{c+1} \leq N \leq 24k^2(2n)^{c+1}$.

3.4 Connecting Resolution Refutations and Cop Strategies

The proof of our width-depth trade-off exploits a connection between the compressed Tseitin formula and the compressed cop-robber game played on the same graph compression, similar to a connection already known for the setting with no compression [GTT20].

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. A similar upper bound was also given in [BBI16, Lemma 56].

Lemma 3.6 (Small-width refutation). *The formula* $\text{Ts}(G_{\text{cyl}})_{\equiv}$ *has a resolution refutation of width* $k + 3$ *and size* $O(k(L + r)2^k)$.

Proof. We will construct a resolution refutation of $\text{Ts}(G_{\text{cyl}})$ of the required width and size. Applying the substitution $x_e \mapsto x_{[e]}$ to all the variables occurring in that refutation yields a refutation of $\text{Ts}(G_{\text{cyl}})_{\equiv}$ in the same width and size.

In fact, we specify the refutation constructed in [BBI16, Lemma 56] for any graph to G_{cyl}, and check that its width is $k + 3$. We start by defining $|G_{\text{cyl}}|$ vertex sets $V_{ij} \subseteq V(G_{\text{cyl}})$ as follows

$$V_{ij} = \{(a, b) \in V(G_{\text{cyl}}) \mid b < j \text{ or } (b = j) \wedge (a \leq i)\}. \tag{3.8}$$

Intuitively, these sets correspond to the parts of the graph searched by the cops when performing a sweep from left to right. We consider the cut sets E_{ij} defined as all the edges in $E(G_{\text{cyl}})$ crossing from V_{ij} to $V(G_{\text{cyl}}) \setminus V_{ij}$, (i.e., edges having one endpoint in V_{ij} and the other in $V(G_{\text{cyl}}) \setminus V_{ij}$).

The refutation of $\text{Ts}(G_{\text{cyl}})$ will derive the parity constraints $\bigoplus_{e \in E_{ij}} x_e \equiv \bigoplus_{v \in V_{ij}} \chi(v)$, denoted $\mathcal{C}_{ij}$, in order increasing in i (modulo k) first, then in j, that is, the sequence of derived constraints is

$$(\mathcal{C}_t)_{t=1}^{|G_{\text{cyl}}|} = (\mathcal{C}_{11}, \mathcal{C}_{21}, \mathcal{C}_{31}, \ldots, \mathcal{C}_{k1}, \mathcal{C}_{12}, \ldots, \mathcal{C}_{22}, \mathcal{C}_{32}, \ldots). \tag{3.9}$$

Let i, j be such that $\mathcal{C}_t = \mathcal{C}_{ij}$. For the axioms at vertex (i, j), denoted A_{ij}, together with $\mathcal{C}_{t-1}$ it holds that $\mathcal{C}_{t-1}, A_{ij} \vDash \mathcal{C}_t$. Hence, resolution can derive all the clauses of $\mathcal{C}_t$ from A_{ij} and $\mathcal{C}_{t-1}$. Furthermore, the width of this derivation is $|E_{n_{i-1}} \cup E_{n_i}| - 1 \leq k + 3$, and its size is at most $\sum_t 8|\mathcal{C}_t| = \text{O}(k(L + r)2^k)$.

Note that the parity constraint $\mathcal{C}_{|G_{\text{cyl}}|-1}$ and the axioms at $(k, L + 2r)$ are contradictory due to the odd charge, so the proof derives $\bot$ in the last step. □

Furthermore, the cops can win the compressed game by playing according to a strategy derived from a refutation of the compressed formula $\text{Ts}(G_{\text{cyl}})_{\equiv}$, as stated below. We prove a more general result, which holds for any Tseitin formula that is nice with respect to a given graph compression.

Lemma 3.7 (Cops simulate refutation). *The following holds for any graph compression* $(G, \equiv_V, \equiv_E)$ *and nice* $\text{Ts}(G)$ *with associated assignment* α_{Ts}. *If there is a width-*w *and depth-*d *resolution refutation of the compressed* $\text{Ts}(G)$, *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. Recall that 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 $\mathrm{Ts}(G)_{\equiv}$, we construct a strategy for the cops based on the DAG G_π associated with π. During every play of the game, the cops trace a path in G_π from the root (associated with the empty clause $\bot$) to a leaf (associated with an axiom of $\mathrm{Ts}(G)_{\equiv}$). During every round of the game, the cops take one step on G_π. Assume that the cops are at the clause D^i in G_π at the beginning of the ith round, that the robber is at v_i, and that the cop position is $\mathbb{C}_i \subseteq V(G)$. The cops maintain that the following invariants hold at the beginning of every round.

(1) Each variable $x_{[e]}$ in D^i is associated to a vertex $v \in \mathbb{C}_i$ that is incident to some edge in class $[e]$;
(2) There is a total assignment α_i on $\{x_{[e]} \mid e \in E(G)\}$ that falsifies only the parity at v_i, and moreover, $\alpha_i(D^i) = 0$.

To show that invariants can be maintained, we proceed by induction on the rounds. Initially, all cops are lifted and $\alpha_1 = \alpha_{\mathrm{Ts}}$, so the invariants hold at the start of the first round.

Assume that the invariants hold at the beginning of the ith round, and that the clauses D^i_1 and D^i_2 are used to derive D^i. Denote by $x_{[e_i]}$ the variable resolved to derive D^i. Since $|\mathbb{C}_i| \leq w + 1$, there is either at least one lifted cop, or there is a vertex in $\mathbb{C}_i$ to which no variable $x_{[e]}$ in D^i is associated, since the width of D^i is at most w. If so, lift that cop. A lifted cop then signals a vertex v that is incident to some edge in class $[e_i]$. The robber does a compressible move M from v_i to some v_{i+1}, and a lifted cop lands at v. We associate $x_{[e_i]}$ to v. We choose the total assignment α_{i+1} to be α_i, with the value of $[e]$ flipped if and only if $e \in M$. This is well-defined since M is closed under $\equiv_E$. Finally, we let D^{i+1} be the clause D^i_j, $j \in \{1, 2\}$, such that α_{i+1} falsifies the literal over $x_{[e_i]}$ in D^i_j.

Let us verify that the invariants hold at the beginning of round $i+1$. Invariant (1) holds by construction. To see why Invariant (2) holds, first note that α_{i+1} falsifies only the parity constraint at $[v_{i+1}]$: By the inductive hypothesis, α_i only falsifies the parity at $[v_i]$, and M flips only the parities at $[v_i]$ and $[v_{i+1}]$ by definition. Furthermore, $\alpha_{i+1}(D^{i+1}) = 0$ because α_{i+1} falsifies the literal over $x_{[e_i]}$ in D^{i+1}, and $\alpha_{i+1} = \alpha_i$ on the rest of the literals in D^{i+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. But, again by the inductive hypothesis on (2), α_i falsifies all these literals.

Finally, we need to show that the cops can win in one round once they reach a leaf in G_π. The clause D associated with this leaf is an axiom, and therefore expanded from the parity constraint at a vertex class. By Invariant (2), the total assignment the cops keep falsifies this axiom and thus the parity at this vertex class. Therefore, 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 Invariant (1), either a cop is occupying some $u' \equiv_V u$ and the cops win, or every neighbor of u has a cop in its class. In the latter case, a lifted cop moves to u in this round while the robber has no compressible move, and the cops win. □

4 Lower Bound for The Compressed Cop-Robber Game

In this section, we prove the lower bound for the compressed game in Theorem 2.2, stated in a more detailed version below. We use the graph compression $(G_{\mathrm{cyl}}, \equiv_V, \equiv_E)$ constructed in Definition 3.2.

Theorem 4.1 (cop-robber (detailed)). *For any $k \geq 3$ and $c \leq k - 1$, we have that $k + 1$ cops can win the compressed game on $(G_{\mathrm{cyl}}, \equiv_V, \equiv_E)$, but as long as there are at most $k + c$ cops, the robber can survive $(L - 2r)/(8(k + c))$ rounds.*

Let us verify that the supercritical width-depth trade-off stated in Theorem 2.5 follows from this theorem, together with the relation between the cop-robber game and resolution width and depth in Lemma 3.7 and the upper bound for resolution in Lemma 3.6.

Proof of Theorem 2.5. Fix $n \in \mathbb{N}$, let $k = k(n)$, $c = c(n)$, and let G_{cyl} be the compressed cylinder with these parameters. We know from Section 3.3 that $\mathrm{Ts}(G_{\mathrm{cyl}})_{\equiv}$ has N variables and $\Theta(N)$ clauses, for $2k^2 n^{c+1} \leq N \leq 24k^2(2n)^{c+1}$. Furthermore, $L \in [2(k + c)n^k, 2^{k+1}(k + c)n^k]$ and $r = k + c + 1$.

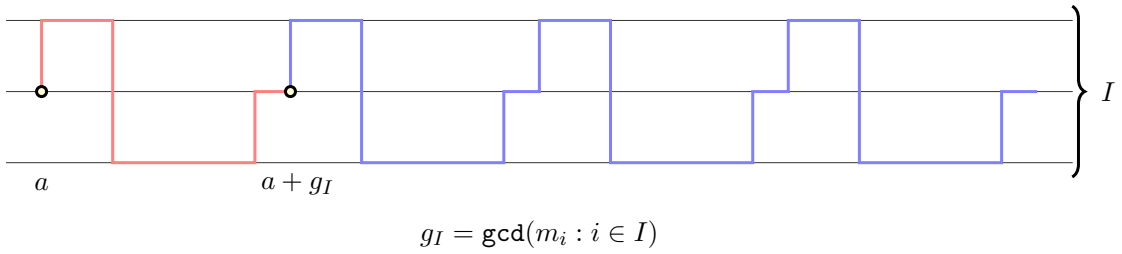


Figure 4: A special move P starting in column a is shown as an I-periodic path, where I is the underlying row set.

Plugging these parameters into Lemma 3.6 yields a resolution refutation in width $k+3$ and size $\mathrm{O}(k^2(4n)^k)$. Together with Lemma 3.7, Theorem 4.1 implies that any resolution refutation in width at most $k+c-1$ must have depth at least $(L-2r)/(8(k+c))-1 = \Omega(n^k)$. □

In Section 4.1, we set up the technical concepts necessary for the proof of Theorem 2.2. 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.

4.1 Preliminary Discussion

Throughout this section, I will be a contiguous row set, that is $I \subseteq [k]$ is a cyclic interval modulo k. 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 $G_{\mathrm{cyl}}|_{I\times[a,b]}$ that can be obtained as follows. Take a simple path P in $G_{\mathrm{cyl}}|_{I\times[a,a+g_I]}$ from column a to column $a+g_I$, starting and ending in the same row. Furthermore, we require that, on columns a and $a+g_I$, P shares no edges, in the following sense: there are no rows $i_1, i_2 \in [k]$ such that $\{(i_1,a),(i_2,a)\} \in P|_{I\times\{a\}}$ and $\{(i_1,a+g_I),(i_2,a+g_I)\} \in P|_{I\times\{a+g_I\}}$. To obtain an I-periodic path, we extend P periodically on $G_{\mathrm{cyl}}|_{I\times[a+g_I,a+2g_I]}$ 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 $|I| = 2$.

Observation 4.3 (Periodic paths give special compressible moves). Suppose W is the set of vertices occupied by cops in the compressed game as described in Definition 3.1. If P is an I-periodic path ($I \neq \emptyset$) from vertex v_1 on column 1 to vertex v_2 on column $2r+L$ and it avoids W, then P is a compressible move. We will call such path a *special move* of the robber from v_1 to v_2, as shown in Figure 4.

Proof. To see why it is a compressible move, note that P is a complete g_I-periodic (horizontally) repetition of some edges in $I \times [1, 1+g_I]$. Recall that horizontal edges are equivalent with period m_i, and vertical edges with period $\gcd(m_i, m_{i+1})$. As $g_I \mid \gcd(m_i, m_{i+1})$ and $g_I \mid m_i$, the edges in P are closed under $\equiv_E$, and thus Condition (G2)a holds. Condition (G2)b, stating that no vertex equivalence class occupied by the cops is touched by P is holds by assumption. Since P is a path connecting v_1 and v_2, which are singleton classes in $\equiv_V$, v_1 and v_2 are the only vertices in G_{cyl} with an odd number of incident edges in P, which is exactly what is required by Condition (G2)c. □

We will also use subsets of the special moves defined above, to be able to move from any column in the left (or right) part of G_{cyl} to any column in the right (or left) part of G_{cyl}. For any two columns a and b in the left and right part of G_{cyl}, respectively, let v_a be the *first* intersection of P with column a and v_b be the *last* intersection of P with column b. We call the subpath of P from v_a to v_b the *(a,b)-truncation of P*. The fact that any (a,b)-truncation of P is also a compressible move follows from the same proof, noting that v_a and v_b are singleton classes in $\equiv_V$.

Next, we want to set up the notions which allow us to analyze the cop position throughout the game. We start by introducing some notation. For $W \subseteq V$ and a row set $I \subseteq [k]$, we write

$$W_I \coloneqq W \cap (I \times [1, 2r + L]) \tag{4.1}$$

for the restriction of W on I. We abbreviate $W_{\{i\}}$ as W_i. Next, we let

$$W_{\equiv_I} \coloneqq \{v \in V \mid (\exists v' \in W_I) \text{ s.t. } v, v' \text{ are on the same row and their distance is 0 mod } g_I\}. \tag{4.2}$$

This 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 if the column b is $W_{\equiv_I}$-free. Furthermore, we write $\operatorname{diam}(W)$ for the difference of the maximum and minimum indices of columns on which W is nonempty. Finally, we use $\operatorname{row}_{\text{unique}}(W) \subseteq [k]$ to denote the set of rows where W has at most one vertex, and call them the *unique rows of* W.

Definition 4.4 (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.

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 $1 \leq |I| \leq a$.

The next fact is immediate from the assumption that $W \subseteq I \times [a, b]$.

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

The following proposition gives some useful properties of separating sets.

Proposition 4.6. *Let $W \subseteq V$ and $\emptyset \neq I \subseteq [k]$ be a cyclic interval such that $|W_I| < g_I$. Then the following hold.*

1. *There is at least one $W_{\equiv_I}$-free column, and $W_{\equiv_I}$-free columns appear with period g_I.*
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 why the first item holds, note that since $\left|W_{\equiv_I} \cap (I \times [1, g_I])\right| = |W_I| < g_I$, 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, we first show the "if"-direction by contrapositive. Assume that W is not I-separating. Then, there is an I-periodic path P from column 1 to column $2r + L$ avoiding $W_{\equiv_I}$. For any good period $I \times [b, b + g_I]$, taking the subpath of P starting from the last vertex of P in column b up to the next vertex of P in column $b + g_I$ shows that $W_{\equiv_I}$ is not a vertex separator of $G_{\text{cyl}}|_{I\times[b,b+g_I]}$.

For the other direction, suppose $W_{\equiv_I}$ is not a vertex separator on a good period $I \times [b, b + g_I]$ of $W_{\equiv_I}$. This means that there is a path $P = (v_1, \ldots, v_t)$ 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 v_1 is the last vertex of P on column b and v_t is the first vertex of P on column $b + g_I$. Then we extend P on column $b + g_I$ to the row of v_1, which is possible since I is a a cyclic interval 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 $2r + L$, while remaining a simple path. It does not intersect with W by the definition of $W_{\equiv_I}$. Thus, this 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 $W_{\equiv_I}$. □

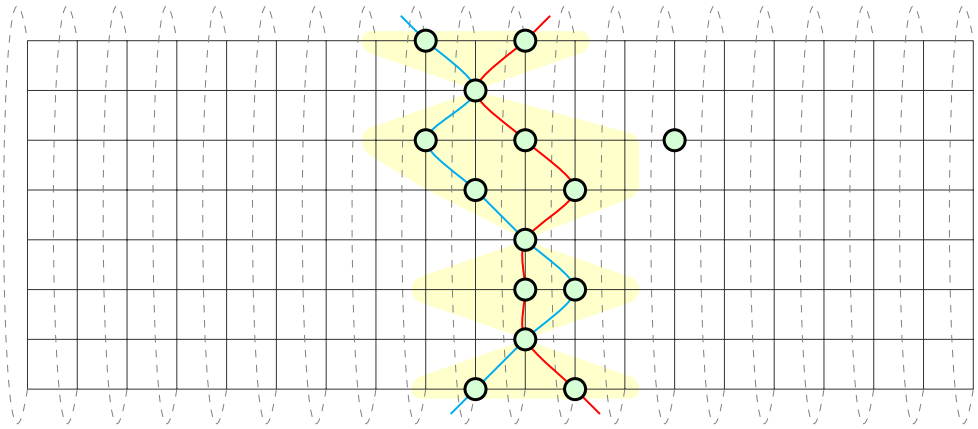


Figure 5: The green circles are vertices in $W_{\equiv}$ (i.e., they are in the equivalent classes of the cops). The red and blue curves illustrate two minimal vertex separators contained in $W_{\equiv}$. The yellow region represents all virtual cordons associated with W.

For the next proposition, recall that the term *minimal* is used with respect to the set-inclusion relation.

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

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 $G_{\mathrm{cyl}}|_{[k]\times[1,c]}$ or $G_{\mathrm{cyl}}|_{[k]\times[c,2r+L]}$Assume without loss of generality that S is a vertex separator for $G_{\mathrm{cyl}}|_{[k]\times[1,c]}$. Then $S_{[k]\times[1,c]}$ is also a vertex separator for G_{cyl}, contradicting the minimality of S. □

Finally, we define the concept of virtual cordons, which is key to our improved analysis over [GLNS23]. Assume that the cops currently occupy all vertices in a set $W \subseteq V$. Roughly speaking, a virtual cordon S of W is a vertex separator that the cops could conceivably occupy starting from this position, thus cordoning the robber off into a small part of the graph.

Definition 4.8 (Virtual cordons). 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 $j \in \mathrm{row}_{\mathrm{unique}}(W)$.

The next definition of a *critical set* combines two properties: First, the ability for the cops, again starting from the set W, to corner the robber using a virtual cordon. Second, the robbers movement is severely limited, because W separates the graph.

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, this means that the robber is in trouble if the cops manage to occupy a critical set with a virtual cordon close to the robber's position. Therefore, the robber strategy will be carefully crafted to let the robber stay away from such critical sets, and monitor their formation closely.

Regarding the definition of critical sets, we note that both conditions in Definition 4.9 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, roughly stating that they cannot be spread far apart if W is small.

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 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

$$|\text{non-unique rows of } W| \leq (k+c) - k = c\,. \tag{4.3}$$

We list $\text{row}_{\text{unique}}(W)$ 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 $\Delta_1 \coloneqq (a_2 - a_1) - (b_2 - b_1)$, 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 $\Delta_j \coloneqq (a_{j+1} - a_j) - (b_{j+1} - b_j)$ 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 $M = \text{lcm}(m_{i_1}, \ldots, m_{i_t})$. 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, 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. □

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 $|W| \leq k + c$. 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 (4.3) we have

$$|\text{non-unique rows of } W| \leq c\,. \tag{4.4}$$

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 (4.4) and since $c \leq k - 1$. 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 $I_1 \coloneqq [i_1, i]$ and $I_2 \coloneqq [i, i_2]$. Then by (4.4) again we have $(i - i_1) + (i_2 - i) \leq c + 1$ (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,

$$|a - b| < k + c\,. \tag{4.5}$$

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 $S_1 \cap (\text{row } i) \neq \emptyset$, 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

$$|a - y| < |S_1| \le |W_{I_1}|\,, \tag{4.6}$$

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

$$|b - z| < |S_2| \le |W_{I_2}|\,. \tag{4.7}$$

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 $I \coloneqq I_1 \cup I_2$. By Property (P1) of the parameters, we have $g_I \ge 2(k + c)$. On the other hand, by (4.5), (4.6), (4.7) we have

$$|y-z| \le |y-a|+|a-b|+|b-z| \le (|W_{I_1}|-1)+(k+c-1)+(|W_{I_2}|-1) < k+c+|W| < 2(k+c)\,. \tag{4.8}$$

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. □

Let us now also consider S_3 which is S restricted to the rows $I_3 \coloneqq [i_2, i_1]$ (a cyclic interval mod k). By Fact 4.5, S_1, S_2 and S_3 are vertex separators for $G_{\text{cyl}}|_{I_1}$, $G_{\text{cyl}}|_{I_2}$ and $G_{\text{cyl}}|_{I_3}$, 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 $G_{\text{cyl}}|_{J_1}$ and T_2 is a vertex separator for $G_{\text{cyl}}|_{J_2}$ 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 $G_{\text{cyl}}|_{J_1 \cup J_2}$.

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 $G_{\text{cyl}}|_{J_1 \cup J_2}$ 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 $G_{\text{cyl}}|_{J_1}$, or

$G_{\text{cyl}}|_{J_2}$, 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 $G_{\text{cyl}}|_{J_1}$ or $G_{\text{cyl}}|_{J_2}$, 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 $G_{\text{cyl}}|_{J_1}$, 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 $G_{\text{cyl}}|_{J_1}$.

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 $G_{\text{cyl}}|_{J_1}$ and $G_{\text{cyl}}|_{J_2}$. 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 $G_{\text{cyl}}|_{J_1}$ and $G_{\text{cyl}}|_{J_2}$ (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. □

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 $v \in W \backslash W^-$ 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 $G_{\text{cyl}}|_I$ such that on each row $j \in I$, $|(S_1)_j| \le |W^-_j|$, 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 $G_{\text{cyl}}|_I$) 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. □

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 $(G_{\text{cyl}}, \equiv_V, \equiv_E)$, where we choose the moduli and parameters in the construction according to Item 3. 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

$$(\emptyset = \mathbb{C}_1^- \to \mathbb{C}_1^+) \to \ldots \to (\mathbb{C}_t^- \to \mathbb{C}_t^+) \to (\mathbb{C}_{t+1}^- \to \mathbb{C}_{t+1}^+) \to \cdots \tag{4.9}$$

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 $G_{\text{cyl}}|_{[k]\times[a_t,a_l]}$. 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 Observation 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 $\left|\text{row}_{\text{unique}}(\mathbb{C}_t^+)\right| > 0$, we have for all $v \in S$ and $v' \in S'$ that

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

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 $|h_1 - a_l|$ or $|h_2 - a_r|$ 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 $\text{row}_{\text{unique}}(\mathbb{C}_t^-) \cap \text{row}_{\text{unique}}(\mathbb{C}_t^+) = \text{row}_{\text{unique}}(\mathbb{C}_t^+)$ 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 (4.10) 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 2.6 (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 of the XOR-substituted formula $\mathrm{XOR}_m(F)$, 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 $\mathsf{fix}(\rho) = \{i \in [n] : \rho(i) \neq \star\}$ and the *width* of ρ to be $|\mathsf{fix}(\rho)|$. 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 2.6 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. □

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 $(x, o) = (\mathrm{XOR}_m^n(y), o) \in \mathcal{S}$. □

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. □

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 $i \in \mathsf{fix}(\rho_{v'})$ (if there is no such i then the claim holds trivially), which implies that $|\{j \mid \rho_{v'}(i,j) = \star\}| = 0$. Let $(\hat{\imath}, \hat{\jmath})$ be the query label of v.

If $i \neq \hat{\imath}$, 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 $i \in \mathsf{fix}(\rho_v)$. We can therefore conclude that

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

Similarly, if $i = \hat{\imath}$, 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

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

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 $\rho_{v'}(i,\hat{\jmath}) = b$ and for all $j \neq \hat{\jmath}$ that $\rho_{v'}(i,j) = \rho_v(i,j)$. □

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 $i \in \mathsf{fix}(\rho_v)$. 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 $\hat{\imath}$ 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 $|\{j \mid \rho_{w'}(\hat{\imath}, j) = \star\}| \geq 2$, otherwise $\sigma_u(\hat{\imath}) \neq \star$ or $|\{j \mid \rho_u(\hat{\imath}, j) = \star\}| \geq 2$, 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 $|\{j \mid \rho_u(\hat{\imath}, j) = \star\}| \leq 1$. 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 $\log s$. □

We conclude this section by noting that, if we increase the threshold $|\{j \mid \rho_{v'}(\hat{\imath}, j) = \star\}| \leq 1$ to $|\{j : \rho_{v'}(\hat{\imath}, j) = \star\}| \leq c$ 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 $\mathcal{S} \circ \mathrm{XOR}_m^n$, 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 2.8 (Restated) (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.*

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 2.8, 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 $t \le r$,

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

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:

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

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 $\forall i \in [r]$,

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

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

Assume $C_{i,y}$ mentions a many y-variables, then $a \ge 1$. 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 $a \cdot \frac{1}{m} \cdot (\frac{1}{2})^{a-1} \le \frac{1}{m}$. To see (6.5), the probability that all of them are set to 0 is $\frac{m-a}{m} \cdot (\frac{1}{2})^{a} \le \frac{1-1/m}{2}$. The claim follows as described. □

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

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

$$\Pr_{\rho}[w(C\restriction_{\rho}) \ge w] \le \left(\frac{2}{m+1}\right)^{w}. \tag{6.6}$$

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

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

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

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

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

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

With Lemma 6.3 at hand, we can finish the proof of Theorem 2.8.

Proof of Theorem 2.8. 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. □

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 2.14 (Restated) (Lifting for triangle-DAGs). *Let F be an unsatisfiable CNF formula over n variables, and let $m, w \in \mathbb{N}, \delta > 0$ be such that $w \leq n$, $0 < \delta < 1 - \frac{1}{\log m}$ and $m \geq (50n/\delta)^{2/\delta}$. 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 2.14 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 2.14 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, $S^X \coloneqq \{x \in [m]^n \mid (\{x\} \times \{0,1\}^{mn}) \cap S \neq \emptyset\}$, and $S^Y \coloneqq \{y \in \{0,1\}^{mn} \mid ([m]^n \times \{y\}) \cap S \neq \emptyset\}$. 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, $1 > \delta > 0$ 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) $H_\infty(\boldsymbol{X}_I) \geq \delta|I| \log m - 1$ *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 $I = \mathsf{fix}(\rho)$, define the width-$|I|$ clause $C_\rho := \bigvee_{i \in I} z_i^{1-\rho(i)}$ where $z^1 := z$ and $z^0 := \overline{z}$. 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 $H_\infty(\boldsymbol{D}_I) \geq \delta|I| \log m - 1$ for all nonempty $I \subseteq [n] \setminus \mathsf{fix}(\alpha)$; and*
2. *$H_\infty(R^Y) \geq mn - m^\delta/(16n) - 1$.*

If for some $\rho \in \{0,1\}^{\mathsf{fix}(\alpha)}$ it holds that $\mathrm{IND}(x_{\mathsf{fix}(\alpha)}, y_{\mathsf{fix}(\alpha)}) = \rho$ 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 $\mathsf{fix}(\alpha) = [n]$, 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 $|\mathsf{fix}(\alpha)| < n$.

Let $D \subseteq (R^X)_\alpha$ be a subset witnessing property 1. Let $J := [n] \setminus \mathsf{fix}(\alpha)$. We consider the rectangle $R' := D_J \times R^Y_J \subseteq [m]^{|J|} \times \{0,1\}^{m|J|}$, where D_J and R^Y_J 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

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

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 $\mathrm{IND}(\{x^{**}\} \times R^Y_J) = \{0,1\}^{|J|}$. The concatenation of α and x^{**} provides the desired element $x^* \in D \subseteq R^X$. □

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 we 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 $R^{\mathrm{in}}_\beta \subseteq R_\beta \cap T$ 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 $X^T_{\text{err}} \subseteq [m]^n, Y^T_{\text{err}} \subseteq \{0,1\}^{mn}$ 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,*

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

- Error. *If x is not secured by any strip in $\mathsf{Strips}(T)$, then $x \in X^T_{\text{err}}$.*
- Maximality. *If there exists a rectangle $R \subseteq T^X \times (T^Y \setminus Y^T_{\text{err}})$ that is α-pre-structured for some pointer α with $|\mathsf{fix}(\alpha)| \leq w$ and $\text{IND}(R) \subseteq C_\beta^{-1}(0)$ 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, $|X^T_{\text{err}}| \leq m^{n-(1-\delta)w}$ and $|Y^T_{\text{err}}| \leq 2^{mn - m^\delta/(16n) + (w+1)\log(2mn)}$.

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 2.14 using the Triangle Lemma and Corollary 7.3.

Proof of Theorem 2.14. Let Π be any triangle DAG of size $m^{(1-\delta)w}/2$ solving $S \circ \text{IND}^n_m$. 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 $T_1, \ldots, T_s$ 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 $X^0_{\text{err}} = Y^0_{\text{err}} := \emptyset$. For $i = 1, \ldots, s$ do the following in order:

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

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

2. Let $X^{T_i}_{\text{err}}$ and $Y^{T_i}_{\text{err}}$ be the X- and Y-error sets, respectively, obtained by applying the Triangle Lemma to T_i.
3. Define $X^i_{\text{err}} := X^{i-1}_{\text{err}} \cup X^{T_i}_{\text{err}}$ and $Y^i_{\text{err}} := Y^{i-1}_{\text{err}} \cup Y^{T_i}_{\text{err}}$.

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 $\bot$ 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 $|\mathsf{fix}(\alpha)| \leq w$ 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 $|\beta^{-1}(0) \cup \beta^{-1}(1)| = |\mathsf{fix}(\alpha)| \leq w$.

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

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

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

$$H_\infty\left(R^X_I\right) \geq |I| \log m - 1 \geq \delta |I| \log m, \tag{7.9}$$

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 $m \geq (50n/\delta)^{\frac{2}{\delta}}$. Therefore, $H_\infty(R^Y \setminus Y^R_{\mathrm{err}}) \geq mn - 1 \geq mn - m^\delta/(16n)$. We therefore conclude that $R^X \times (R^Y \setminus Y^R_{\mathrm{err}})$ is a $\star^n$-(pre-)structured rectangle.

By the maximality condition of Triangle Lemma applied to $R^X \times (R^Y \setminus Y^R_{\mathrm{err}})$, 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 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 $\mathrm{IND}(R_\beta) = \{0,1\}^n = \bot^{-1}(0)$. Therefore $\bot \in \mathcal{C}(R)$.

– *Leaves.* Consider any leaf triangle T of Π. By definition, there is an axiom clause $C_i \in F$ such that $\mathrm{IND}(T) \subseteq C_i^{-1}(0)$. Therefore, for any clause $C_\beta \in \mathcal{C}(T)$, we have $C_\beta^{-1}(0) = \mathrm{IND}(R_\beta) = \mathrm{IND}(R^{\mathrm{in}}_\beta) \subseteq \mathrm{IND}(T) \subseteq C_i^{-1}(0)$, 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 $R_\beta = S \times Y^T_\beta$ 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 $R^{\mathrm{in}}_\beta \subseteq R_\beta \cap T$. Since T is covered by its children T_1 and T_2,

$$R^{\mathrm{in}}_\beta \subseteq T \subseteq T_1 \cup T_2. \tag{7.11}$$

We claim that at least one of T_1 or T_2 contains a sub-rectangle $Q = Q^X \times Q^Y \subseteq R^{\mathrm{in}}_\beta$ with X- and Y-density at least half that of R^{in}_β. To see this, order the rows/columns according to the ordering of T_1, then the center p of R^{in}_β divides R^{in}_β into four quadrants. If $p \in T_1$ then, as T_1 is a triangle, the top-left quadrant Q of R^{in}_β 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, $H_\infty(Q^Y) \geq H_\infty(Y^T_\beta) - 1$ and $H_\infty(Q^X_J) \geq H_\infty(S_J) - 1$ 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 $Q \subseteq T_1$.

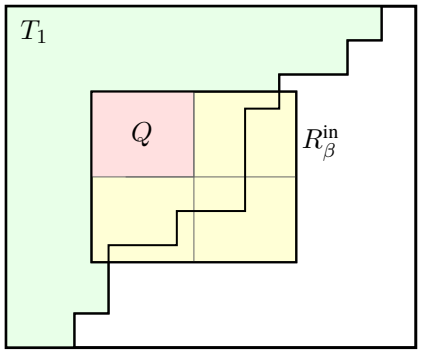


Figure 6: The structured rectangle R^{in}_β for triangle T, whose quadrant Q is contained entirely within child T_1.

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 $X^{T_1}_{\text{err}}$ and $Y^{T_1}_{\text{err}}$ from T in the Error Removal step, $x^* \notin X^{T_1}_{\text{err}}$ and $Q^Y \subseteq T^Y$ is disjoint from $Y^{T_1}_{\text{err}}$. Thus, x^* is secured by a strip S' of T_1 defined by some pointer α'. By the Triangle Lemma,

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

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. □

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 $\text{Strips}(T)$ and error sets which satisfy the *security*, *error* and *maximality* properties of the lemma.

Let parameters w and δ be given. Given a triangle T, for every pointer α with $|\text{fix}(\alpha)| \leq w$ 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 for T as follows. As a row set, S is equal to $(T^X)_\alpha$.

- *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 $\text{fix}(\beta) = \text{fix}(\alpha)$ consider the set of columns

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

  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^S_{err}.

– *Inner Rectangle.* It remains to show that we can find some sub-rectangle $R^{\text{in}}_\beta \subseteq R_\beta \cap T$ 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 $R^{\text{in}}_\beta := S' \times Y_\beta$ is only above (and including) $\{x^S\} \times Y_\beta \subseteq T$. Hence, as T is a triangle, $R^{\text{in}}_\beta \subseteq T$. Finally, note that as R_β was not categorized as "error", R^{in}_β 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 $X^T_{\text{err}} \subseteq [m]^n$ and $Y^T_{\text{err}} \subseteq \{0,1\}^{mn}$ as follows:

– *X-Error.* Let X^T_{err} be the set of rows in T^X which are not secured by *any* strip in $\mathsf{Strips}(T)$.
– *Y-Error.* Let Y^T_{err} be collected over all strips $S \in \mathsf{Strips}(T)$, that is, $Y^T_{\text{err}} := \bigcup_{S \in \mathsf{Strips}(T)} Y^S_{\text{err}}$.

A depiction of a strip is in Figure 7.

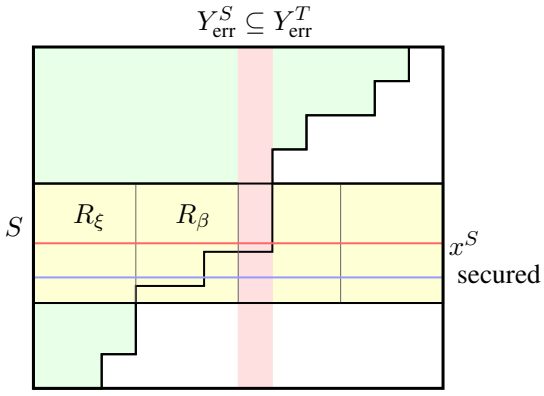


Figure 7: A strip S of a triangle, including two pre-structured rectangles R_ξ, R_β, a set of error columns Y^S_{err}, and an example of a secured row.

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,

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

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,

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

Now, for the maximality property, assume there is a rectangle $R \subseteq T^X \times (T^Y \setminus Y^T_{\text{err}})$ that is α-pre-structured for some pointer α with $|\mathsf{fix}(\alpha)| \leq w$ and $\text{IND}(R) \subseteq C^{-1}_\beta(0)$ for some $\beta \in \{0,1,\star\}^n$ with $\mathsf{fix}(\beta) = \mathsf{fix}(\alpha)$. 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 $\text{IND}(R) \subseteq C^{-1}_\beta(0)$ it follows that R^Y is a subset of $Y_\beta = \{y \in \{0,1\}^{mn} \mid \text{IND}(\alpha_{\mathsf{fix}(\alpha)}, y_{\mathsf{fix}(\alpha)}) = \beta_{\mathsf{fix}(\alpha)}\}$ defined in (7.14). As $R^Y \cap Y^T_{\text{err}} = \emptyset$ and $R^Y \subseteq Y_\beta$ (and $R^Y \neq \emptyset$ since R is α-pre-structured), it must be the case that $Y_\beta \not\subseteq Y^T_{\text{err}}$ 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^T_{err} in $[m]^n$ is less than $m^{-(1-\delta)w}$.

Proof. Suppose for contradiction that X^T_{err} has density at least $m^{-(1-\delta)w}$. For simplicity, we denote $\widehat{X} \coloneqq X^T_{\text{err}}$. 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 $\mathsf{fix}(\alpha) = I$ that witnesses $\Pr[\widehat{\mathbf{X}}_I = \alpha_I] \geq m^{-\delta|I|}$. If no such I exists, we let $I \coloneqq \emptyset$ and $\alpha = \star^n$. We record the following two basic properties:

(1) $|I| \leq w$,

(2) $\widehat{X}_\alpha \coloneqq \{x \in X^T_{\text{err}} \mid x_I = \alpha_I\}$ 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 $\mathsf{fix}(\alpha') = J$ such that $\Pr_{x \sim \widehat{X}_\alpha}[x_J = \alpha'_J] \geq m^{-\delta|J|}$. Let $\alpha \circ \alpha'$ be the pointer with $\mathsf{fix}(\alpha \circ \alpha') = I \cup J$ such that $(\alpha \circ \alpha')_I = \alpha_I$ and $(\alpha \circ \alpha')_J = \alpha_J$. Then

$$\begin{aligned}
\Pr_{x \sim \widehat{X}}[x_{I \cup J} = (\alpha \circ \alpha')_{I \cup J}] &= \Pr_{x \sim \widehat{X}}[x_I = \alpha_I] \cdot \Pr_{x \sim \widehat{X}}[x_J = \alpha'_J \mid x_I = \alpha_I] && (7.18)\\
&= \Pr_{x \sim \widehat{X}}[x_I = \alpha_I] \cdot \Pr_{x \sim \widehat{X}_\alpha}[x_J = \alpha'_J] && (7.19)\\
&\geq m^{-\delta(|I|+|J|)}, && (7.20)
\end{aligned}$$

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 $x \in \widehat{X}_\alpha \subseteq X^T_{\text{err}}$ where X^T_{err} contains only rows of T that are not secured by any strip in $\mathsf{Strips}(T)$. □

Claim 7.6. For any triangle T, the density of Y^T_{err} 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 $|\mathsf{fix}(\alpha)| \leq w$, 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 $2^{|\mathsf{fix}(\alpha)|} \leq 2^w$ many $\beta \in \{0,1,\star\}^n$ with $\mathsf{fix}(\beta) = \mathsf{fix}(\alpha)$, 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, $|Y^T_{\text{err}}| < (2mn)^{w+1} 2^{mn-m^\delta/(16n)}$, and so the density of Y^T_{err} can be upper bounded by $2^{-m^\delta/(16n)+(w+1)\log(2mn)}$. □

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. It is now known that the perfect matching problem is not such an example for circuits [CGR+26]. However, it is still open whether Tseitin formulas are a candidate formula for supercritical trade-offs for cutting planes. If this were the case, this 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

The authors wish to thank Christoph Berkholz, Jonas Conneryd, Daniel Neuen, and Alexander Razborov for helpful discussions. 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. We also gratefully acknowledge that we have benefited greatly from being part of the Basic Algorithms Research Centre (BARC) environment financed by the Villum Investigator grant 54451.

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*.

[BHP10] Paul Beame, Trinh Huynh, and Toniann Pitassi. Hardness amplification in proof complexity. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC '10)*, pages 87–96, June 2010.

[BLV25] Christoph Berkholz, Moritz Lichter, and Harry Vinall-Smeeth. Supercritical size-width tree-like resolution trade-offs for graph isomorphism. In *Proceedings of the 50th International Symposium on Mathematical Foundations of Computer Science (MFCS '25)*, volume 345 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 18:1–18:19, August 2025.

[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.

[BPR97] María Bonet, Toniann Pitassi, and Ran Raz. Lower bounds for cutting planes proofs with small coefficients. *Journal of Symbolic Logic*, 62(3):708–728, September 1997. Preliminary version in *STOC '95*.

[BT26] Sam Buss and Neil Thapen. A simple supercritical tradeoff between size and height in resolution. *Information Processing Letters*, 191:106589, January 2026.

[CCT87] William Cook, Collette Rene Coullard, and György Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, November 1987.

[CD24] Arkadev Chattopadhyay and Pavel Dvořák. Super-critical trade-offs in resolution over parities via lifting. In *Proceedings of the 40th Computational Complexity Conference (CCC '25)*, volume 339 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:19, August 2024.

[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*.

[CGR+26] Bruno Cavalar, Mika Göös, Artur Riazanov, Anastasia Sofronova, and Dmitry Sokolov. Monotone circuit complexity of matching. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing (STOC '26)*, page 1128–1132, June 2026.

[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*.

[GLN25] Martin Grohe, Moritz Lichter, and Daniel Neuen. The iteration number of the Weisfeiler-Leman algorithm. *ACM Transactions on Computational Logic*, 26(1):6:1–6:31, January 2025. Preliminary version in *LICS '23*.

[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.

[GMRS25] Mika Göös, Gilbert Maystre, Kilian Risse, and Dmitry Sokolov. Supercritical tradeoffs for monotone circuits. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC '25)*, pages 1359–1370, June 2025.

[GP18] Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. *SIAM Journal on Computing*, 47(5):1778–1806, October 2018. Preliminary version in *STOC '14*.

[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*.

[HN12] Trinh Huynh and Jakob Nordström. On the virtue of succinct proofs: Amplifying communication complexity hardness to time-space trade-offs in proof complexity (Extended abstract). In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC '12)*, pages 233–248, May 2012.

[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.

[HP18] Pavel Hrubeš and Pavel Pudlák. A note on monotone real circuits. *Information Processing Letters*, 131:15–19, March 2018.

[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.

[Kra98] Jan Krajíček. Interpolation by a game. *Mathematical Logic Quarterly*, 44(4):450–458, 1998.

[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.

[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.

[Oli15] Igor Carboni Oliveira. *Unconditional lower bounds in complexity theory*. Columbia University, 2015.

[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.

[RM99] Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, March 1999. Preliminary version in *FOCS '97*.

[Rob18] Robert Robere. *Unified Lower Bounds for Monotone Computation*. PhD thesis, University of Toronto, 2018. Available at https://hdl.handle.net/1807/92007.

[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 2.3) 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 $\vec{a} = (a_1, \ldots, a_d) \in \mathbb{F}_2^d$ 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 *compressible 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 $(u, v) \in T \Leftrightarrow (u', v') \in T$.

[3] For G_{cyl} with the lexicographical vertex order on $[k] \times [2r + L]$ and the vertex equivalence relation $\equiv_V$ in Definition 3.2, this condition is satisfied.

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 2.1, which is the same as $\equiv_E$ in Definition 3.2.

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 2.1, the *compressed k-cop-edge-robber* game proceeds as in Definition 3.1, 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 $(G_{\text{cyl}}, \equiv_V, \equiv_E)$ in Theorem 4.1. For ease of notation, we will call the game in Definition 3.1 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 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 $(G_{\text{cyl}}, \equiv_V, \equiv_E)$, 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 cop is picked up, and a destination $x \in V(G)$ selected). In the k-VR, the same cop 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 cop.

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 occupied by a cop 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 cop 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 2.3. 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 Item 3. 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 Item 3 are satisfied. We apply the CFI construction with $f, g\colon E(G_{\text{cyl}}) \to \mathbb{F}_2$, where f is the all-zero function and g is 1 only on one edge adjacent to vertex $(1,1)$. This gives us graphs $G_N := \text{CFI}(G_{\text{cyl}}, f)/\equiv_V$ and $H_N := \text{CFI}(G_{\text{cyl}}, g)/\equiv_V$, where N denotes the vertex set size of both. We observe that $N \le 2^{4-1}\left(2r + k^2(2n)^{c+1}\right) < 2^{c+5}k^2n^{c+1}$.

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 $(L-2r)/(8(k+c)) \ge n^k/(32k)$ 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 $m^\delta \ge \frac{4}{\ln 2} n$,*

(1) $H_\infty(\boldsymbol{X}_I) \ge \delta|I|\log m - 1$ for any nonempty $I \subseteq [n]$,
(2) $H_\infty(Y) \ge mn - m^\delta/(8n)$.

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

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 $(S_1, \dots, S_l)$ 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 $(S_1, \dots, S_l)$ 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*

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

In particular, if $p\kappa \ge (r-1)/\ln 2$, 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

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

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

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

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

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

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

Remark B.2. The probability bound in (B.1) was improved by [Rao20, Lemma 4] to $r \cdot \exp(-p\kappa/C_{\text{abs}})$ 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:

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

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 all of 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 $p^0 := \overline{p}$. 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 $|\text{sat}(F)| \le |\text{sat}(F^{mon})|$, where $\text{sat}(\cdot)$ means the set of satisfying assignments.*

To see this, note that F^{mon} can be obtained from a sequence $F_0 := F, F_1, \dots, F_N = F^{mon}$ where F_i monotonize F_{i-1} at the ith variable, and we only need to show that $|\text{sat}(F_{i-1})| \le |\text{sat}(F_i)|$. 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 $P_y \subseteq \text{sat}(F_{i-1})$ then the partial assignment y already satisfies each clause in F_{i-1}, so $P_y \subseteq \text{sat}(F_i)$; if exactly one of $(y,0)$ and $(y,1)$ satisfies F_{i-1}, then $(y,0)$ satisfies F_i. Thus for each y, $|\text{sat}(F_{i-1}) \cap P_y| \le |\text{sat}(F_i) \cap P_y|$. Consequently, $|\text{sat}(F_{i-1})| \le |\text{sat}(F_i)|$, 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 $\kappa \ge 1$, $\underline{A} = \{\underline{a} \mid a \in A\}$ is κ-spread if and only if $H_\infty(\mathbf{A}_I) \ge |I| \log \kappa$ 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. □

Paper B

Average-Case Hardness of Binary-Encoded Clique in Proof and Communication Complexity

Average-Case Hardness of Binary-Encoded Clique in Proof and Communication Complexity

Susanna F. de Rezende
Lund University

David Engström
Lund University

Yassine Ghannane
University of Copenhagen
and Lund University

Duri Andrea Janett
University of Copenhagen
and Lund University

Artur Riazanov
EPFL

Abstract

We study the average-case hardness of establishing that a graph does not have a large clique in both proof and communication complexity. We show exponential lower bounds on the length of cutting planes and bounded-depth resolution over parities refutations of the binary encoding of clique formulas on randomly sampled dense graphs. Moreover, we show that the randomized communication complexity of finding a falsified clause in these formulas is polynomial.

1 Introduction

The k-Clique problem asks whether a given graph G contains a complete subgraph of size k. This is one of the classical **NP**-hard problems [36]. Under the *Exponential Time Hypothesis*, solving k-Clique on an N-vertex graph requires time $N^{\Omega(k)}$ [15], nearly matching the $O(N^k)$ upper bound that can be obtained by simply checking all $\binom{N}{k}$ possible subgraphs. Furthermore, it is **NP**-hard to approximate the maximum size of a clique even within a factor $N^{1-\varepsilon}$ [29, 53].

Showing unconditional super-polynomial lower bounds for k-Clique seems beyond the reach of current techniques, as it would immediately imply $\mathbf{P} \neq \mathbf{NP}$. Thus, significant effort has been made to provide unconditional *evidence* for the hardness of k-Clique. The most direct evidence is hardness in restricted computational models. In circuit complexity, k-Clique was the first problem proven to require exponential-size *monotone Boolean circuits* [45]. In a later breakthrough, Rossman [47, 48] showed that the k-Clique problem is hard for monotone and bounded-depth circuits even *on average*.

1.1 Clique in Proof Complexity

Another type of evidence for hardness comes from *propositional proof complexity*, the study of the complexity of certifying, in various formal proof systems, that a given Boolean formula is a tautology or that it is unsatisfiable.

Proof complexity provides a perspective to analyze several algorithmic approaches to the k-Clique problem: If an algorithm rejects a k-Clique instance, the trace of its execution can be viewed as a certificate of the claim that the given graph *does not* contain a k-clique. This claim can be encoded as a CNF formula asserting that the graph *does* have a k-clique, and in this paper we study certificates of unsatisfiability, also called *refutations*, of this formula. Many classes of algorithms can be captured by propositional proof systems in this way: the core reasoning of state-of-the-art SAT-solvers is captured by *resolution* [4]; Gröebner basis algorithms by *polynomial calculus* [17]; the cutting planes methods in integer linear optimization by the *cutting planes* proof system [18]; and the Lassere hierarchy in semi-definite programming [39] by the *sum-of-squares* proof system.

Assuming the *Nondeterministic Exponential Time Hypothesis* [13], in *every* proof system the shortest refutation of k-Clique for *some* graph has length at least $N^{\Omega(k)}$. On the other hand, unconditional lower bounds are only known for very restricted proof systems. Two encodings of the clique problem into CNF formulas are studied in the literature. The *unary encoding* uses N propositional variables to encode the ith vertex of the purported k-clique, for every i, while the more succinct *binary encoding* uses only $\log N$ variables to encode the same information in binary.

The following results are known about the unary encoding. For *treelike* resolution, there are $N^{\Omega(k)}$ length lower bounds even on average with respect to the Erdős-Rényi graph distribution [7, 40]. The current best known results are analogous average-case bounds for *regular* resolution [2], and Sherali–Adams with bounded coefficients [20]. For general resolution, in the parameter range $N^{5/6} \ll k \leq N/3$, an $\exp(N^{\Omega(1)})$ average-case lower bound for dense graphs is known [3], and when $k < N^{1/3}$, an average-case lower bound $2^{k^{1-\varepsilon}}$ holds [43]. Improving this to an optimal $N^{\Omega(k)}$ lower bounds is a long-standing open problem, see, e.g., [8, 11], even for worst-case instances. In particular, all known techniques for proving resolution lower bounds seem to fail for this task [2].

Weak proof systems can, however, be very sensitive to the choice of encoding, as was shown in, e.g., [19]. This is also the case for the binary encoding of clique, where strong lower bounds are known for resolution: There is an $N^{\Omega(k)}$ average-case lower bound for resolution [41], which was further extended to s-DNF resolution, for $s = o(\log \log n)$, at the cost of some loss in the exponent

of the bound depending on s [19].

1.2 Our contributions

We prove new lower bounds for k-Clique in three settings. Our first result is that the binary encoding of k-Clique requires exponential proof length in cutting planes. Prior to our work, nothing was known about the hardness of k-Clique for cutting planes, even in the treelike setting.

Theorem 1.1 (Informal)**.** *For N-vertex graphs G sampled from the Erdős-Rényi distribution with appropriate edge density, any cutting planes refutation of the binary-encoded clique formula must have length at least $2^{N^{\Omega(1)}}$.*

We consider next the frontier proof system *resolution over parities* (Res($\oplus$)) [34], which recently received a lot of attention [10, 1, 22, 9, 12], particularly following the breakthrough of [21]. Lower bounds are only known for restricted versions of this system, and in particular, nothing is known about the hardness of refuting k-Clique in Res($\oplus$). We prove the following length lower bound on Res($\oplus$) proofs of bounded depth.

Theorem 1.2 (Informal)**.** *For N-vertex graphs G sampled from the Erdős-Rényi distribution with appropriate edge density, any Res($\oplus$) refutation of the binary-encoded clique formula in depth $m^{1.5-\varepsilon}$, where m is the number of variables in the formula, must have length at least $2^{N^{\Omega(1)}}$.*

Lastly, we study the communication version of the problem. Two parties are both given partial information about the members of a purported k-clique in a graph G. Again, G does not actually have a k-clique, and the parties must find a missing edge in the purported clique, while communicating as few bits as possible. This corresponds to finding a falsified clause in the binary-encoded clique formula, if the partial information consists of the bits encoding the purported clique members. We prove the following lower bound in the setting where the parties have access to shared randomness.

Theorem 1.3 (Informal)**.** *For N-vertex graphs G sampled from the Erdős-Rényi distribution with appropriate edge density, the randomized communication cost of finding a falsified clause in the binary-encoded clique formula is $N^{\Omega(1)}$.*

Randomized communication complexity remains the most versatile tool for obtaining lower bounds for *treelike* cutting planes. Most of the known lower bounds are obtained via this connection [32, 5, 31, 27, 33]. Treelike cutting planes is a natural next candidate for proving clique lower bounds in the *unary* encoding.

There is, however, growing evidence that the randomized communication model is too strong to give strong lower bounds for k-Clique. If, for example, the density of Erdős-Rényi graph is constant, then Alice and Bob can randomly sample a pair of indices of their nodes and verify whether or not the corresponding pair is connected with an edge. Thus, we can only hope for a non-trivial lower bound for the dense setting of k-Clique. Another weakness of the communication approach to k-Clique lower bounds was established by Jukna [35], who observed that for a *bipartite* version of the unary k-Clique, the corresponding problem has a $O(\log n)$-cost *deterministic* communication protocol.

Given these drawbacks, we view Theorem 1.3 as an indication of some hope for the communication approach to k-Clique lower bounds: at least in the dense binary setting the barriers above do not apply. Moreover, Theorem 1.3 (qualitatively) generalizes the main theorem in [51] with a much simpler proof.

1.3 Discussion

Our contributions can be viewed as matching the lower bounds known for the *weak* version of *binary pigeonhole principle* (BPHP) with bounds for the binary encoding of k-Clique. In fact, all three bounds are obtained by adapting the techniques previously used to prove lower bounds for the weak binary pigeonhole principle [6, 12], or by adapting extensions of such techniques [24, 51]. Is there a more general principle behind our results, i.e., is there an explicit reduction from BPHP to average-case k-Clique? We know such a reduction exists in the *worst case*—indeed, the BPHP is a special case of the binary-encoded k-Clique—but in the average case such reductions are not known.

One concrete direction would be to prove lower bounds on k-Clique in the treelike version of Th(2), the generalization of cutting planes refutations to degree-2 inequalities. Only the *strong* BPHP is known to be hard for this system [33], and the proof of this fact uses global symmetries of the formula. Therefore, it is unclear how to apply the technique to k-Clique for an Erdős-Rényi random graph, since the latter is unlikely to have symmetries. Does it fail because of the particular technique, or is the strong version of BPHP crucially easier than the weak one?

Open Problem 1.4. Suppose weak BPHP requires exponential tree-like Th(2) refutation. Show that the binary encoding of k-Clique does as well.

Another interesting direction to extend our lower bounds is to the *Ramsey principles* [41]. A graph is *c-Ramsey* if it has neither a clique nor an independent set of size $c \log n$. The Ramsey principle formulas defined in [41] encodes in binary the claim that a given graph is c-Ramsey. For resolution, an asymptotically optimal $n^{\Omega(\log n)}$ length lower bound was proven in [41]. We ask the same question for cutting planes and Res($\oplus$).

Open Problem 1.5. Prove that the Ramsey principles are hard for cutting planes or bounded-depth Res($\oplus$).

In particular, matching the bounds known for resolution would require improving our lower bounds and extending them to randomly sampled graphs for a broader range of parameters, specifically to graphs which are not as dense.

As discussed above, for sparse graphs there is always a *randomized* communication protocol solving k-Clique in binary or unary encoding. However, we can weaken the communication model so it still captures tree-like cutting planes. One such model is a *deterministic* communication protocol with a Greater Than oracle, i.e., the players can compare real numbers at unit cost.

This model is shown to be significantly weaker than randomized communication [14, 16, 25]. Can we still solve k-Clique over a sparse graph?

Open Problem 1.6. Suppose that Π is a deterministic communication protocol with a Greater Than oracle. Can Π solve k-Clique for Erdős-Rényi graphs of constant sparsity with $O(\log n)$ cost?

1.4 Techniques

We give a brief technical overview of our proofs. Notably, all of our techniques are to some extent based on previous work proving lower bounds on BPHP.

Cutting Planes. Theorem 1.1 is proved via a *bottleneck counting* argument [28]. The approach is based on an adaptation of this framework to cutting planes due to Sokolov [50], that was also subsequently refined by Beame and Whitmeyer [6] to prove cutting planes BPHP lower bounds. It leverages a connection between cutting planes proofs and triangle-DAG protocols, saying that small cutting planes refutations of some formula imply small protocols for a related search problem. Informally, the main idea is to then define a partial function μ that maps inputs of the search problem to nodes of the protocol. Intuitively, μ measures progress made by the protocol at some step. By giving a lower bound on the size of the domain of μ and an upper bound on the maximum number of inputs mapped to each node, we establish a size lower bound on the protocol and thus on cutting planes refutations of our starting formula. The twist of our approach is to show that a certain combinatorial property of graphs related to the density of neighborhoods is sufficient to construct such a partial map μ. In particular, dense random graphs asymptotically almost surely exhibit such properties.

Bounded depth Res($\oplus$)**.** Recent progress has led to a BPHP lower bound for depth-$m^{1.5-\varepsilon}$ Res($\oplus$) [12], based on a notion of *closure* for a set of linear forms [21], and the *random walks with restarts* framework [1]. This framework works by performing a random walk on the proof in a top-down manner. If one can bound below the probability that this walk proceeds for many steps without reaching a falsified clause, and assuming that the given refutation is small, some node in the proof must be visited by many random walks, and thus the rank of the linear system at that node must be low. Using the notion of closure, one can then fix only few variables to satisfy that system, which allows repeating this procedure, walking further down the proof, and eventually establishing a depth lower bound.

Adapting the framework to binary-encoded clique formulas, finding a falsified clause is the same as identifying a non-edge in the underlying graph. Our random walk then mimics a uniformly drawn input while maintaining a partial assignment of the clique members to vertices without non-edges between them. We ensure this by always staying within the common neighborhood of previously assigned vertices, and find that the walk can proceed for many steps with sufficient probability. Additionally, we use concentrations bounds on the number of common neighbors of a set of vertices to show that, in the right parameter regimes, after fixing variables, the remaining instance is similar enough to a smaller instance of the original problem, thus allowing us to restart the random walk.

Randomized communication. Theorem 1.3 (qualitatively) generalizes the BPHP lower bound of Yang and Zhang [51]. They used query-to-communication lifting machinery from [26] to essentially reduce the communication lower bound to a corresponding *decision tree* lower bound. The proof in [51] is quite involved due to a white-box use of the lifting machinery. Later, [24] proved a structural theorem that allows a black-box proof of the main result in [51], and this theorem was recently extended and used in [46] to get a lower bound for the clause-search problem for random $O(\log n)$-CNFs. We show how to apply this framework to the clique problem.

2 Preliminaries

We use the symbol $\sqcup$ to denote a disjoint union of sets. We write $\{1, \ldots, n\}$ as $[n]$, where $n \in \mathbb{N}^+$. For a bit string $y \in \{0,1\}^k$, we write $y^{\oplus j}$ for the same bit string with the jth coordinate flipped. Let $\boldsymbol{x} \sim A$ denote $\boldsymbol{x}$ uniformly sampled from a set A. We say that an event occurs *asymptotically*

almost surely (a.a.s) if it occurs with probability $1-o(1)$ as $n \to \infty$. We will also need the following special case of the Chernoff bound:

Theorem 2.1 (Multiplicative Chernoff bounds)**.** *Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ be independent $\{0,1\}$-valued random variables, and let $\boldsymbol{X} = \sum_{i=1}^{n} \boldsymbol{X}_i$, $\mu = \mathbb{E}[\boldsymbol{X}]$. Then, for $0 \leq \delta \leq 1$,*

$$\Pr\left[\boldsymbol{X} \geq (1+\delta)\mu\right] \leq \exp\left(-\frac{\mu\delta^2}{3}\right),$$
$$\Pr\left[\boldsymbol{X} \leq (1-\delta)\mu\right] \leq \exp\left(-\frac{\mu\delta^2}{2}\right).$$

2.1 Graph Theory

Let G be a graph over k blocks $V(G) = V_1 \sqcup \cdots \sqcup V_k$, with $|V_i| = n$ for all $i \in [k]$. In everything that follows, let us assume, without loss of generality, that n is a power of 2. For $u \in V(G)$, let $B(u) \in [k]$ denote the index of the block that contains u, and for every set of vertices $U \subseteq V(G)$, we define $B(U) \coloneqq \{B(u) \mid u \in U\}$. For a subset $U \subseteq V(G)$ we use $N^{\cap}(U, i)$ to denote the *common neighborhood* of U at block i. For a set of blocks $I \subset [k]$, we then use $N^{\cap}(U, I) = \bigcup_{i \in I} N^{\cap}(U, i)$. For all $i \in [k]$, every $u \in V_i$ can be identified with an integer in $[n]$, and we denote its binary representation by $\mathsf{bin}(u) = u_1 \ldots u_{\log(n)}$.

The natural distribution for which k-Clique is conjectured to be hard is an Erdős-Rényi distribution. The p-biased Erdős-Rényi distribution $\mathcal{G}(n,p)$ is defined by sampling an n-vertex graph with each of the $\binom{n}{2}$ edges present with probability p independently of the other edges. The parameter p that conjecturally makes finding a k-clique in $\boldsymbol{G} \sim \mathcal{G}(n,p)$ hard is at the threshold p' where for a lower $p < p'$ the graph $\boldsymbol{G}$ does not have a k-clique a.a.s., and for a higher $p > p'$ it does have a k-clique a.a.s.. By looking at the expected number of k-cliques in $\boldsymbol{G}$, one can conclude that the threshold value is $\Theta(n^{-2/(k-1)})$.

We also define the *k-partite* version of the same distribution: For a fixed partition of the nodes, $\mathcal{G}(n,p,k)$ is sampled from by first sampling $\boldsymbol{G} \sim \mathcal{G}(nk,p)$ and then intersecting $\boldsymbol{G}$ with a complete k-partite graph with n nodes in each part.

2.2 Proof Complexity

We next recall some basic notions from proof complexity; see, e.g., [37, 11] for a more thorough exposition. A Boolean variable x or its negation $\overline{x}$ is called a *literal*, and a disjunction of literals over pairwise disjoint variables $C = \ell_1 \vee \cdots \vee \ell_k$ is called a *clause*. A *CNF formula* is a conjunction of clauses $F = C_1 \wedge \cdots \wedge C_m$. We sometimes call the clauses of F *axioms*, and denote the set of variables occurring in F as $\mathrm{Vars}(F)$.

Cutting Planes. The *cutting planes* proof system [18] operates on systems of linear inequalities. In its syntactic form, it derives new constraints from previously derived ones by *linear combination* and *division with rounding*. Since our lower bounds also apply to it, we define the stronger *semantic* version of the proof system, which subsumes all usual cutting planes derivation rules.

Let $Ax \geq b$ be a system of linear inequalities, where $A \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}^m$. A *semantic cutting planes refutation* of $Ax \geq b$ is a sequence of linear inequalities $\{d_i x \geq c_i\}_{i \in [L]}$, where $d_i \in \mathbb{R}^n$ and $c_i \in \mathbb{R}$, such that the following hold: The final inequality is the trivially false $0 \geq 1$, and for every

$i \in [L]$, the inequality $d_i x \geq c_i$ is part of the linear system $Ax \geq b$, or there are $j, k < i$ such that $d_i x \geq c_i$ follows from $d_j x \geq c_j$ and $d_k x \geq c_k$ by *semantic deduction*. That is, from inequalities $d_j x \geq c_j$ and $d_k x \geq c_k$, one can derive any inequality $d_i x \geq c_i$ for which it holds that any $x \in \{0,1\}^n$ satisfying both $d_j x \geq c_j$ and $d_k x \geq c_k$ satisfies $d_i x \geq c_i$. The *length* of a semantic cutting planes refutation is the number L of inequalities appearing in the sequence.

The following translation enables us to consider cutting planes refutations of unsatisfiable CNF formulas. For every propositional variable x, we add the constraints $0 \leq x$ and $x \leq 1$. For a clause $C = \bigvee_{i \in I} x_i \vee \bigvee_{j \in J} \overline{x_j}$, we add the inequality $\sum_{i \in I} x_i + \sum_{j \in J} (1 - x_j) \geq 1$.

Resolution over Parities. The *resolution over parities* proof system [34], which we refer to as Res($\oplus$), operates on linear equations over $\mathbb{F}_2$. Let $x_1, \ldots, x_n$ be variables taking values in $\mathbb{F}_2$. A *linear form* is a polynomial $\sum_{i=1}^{n} a_i x_i$, where all $a_i \in \mathbb{F}_2$. Taking a linear form f and $a \in \mathbb{F}_2$, we get a *linear equality* $f = a$. A disjunction of linear equations $C = \bigvee_{i=1}^{m} (f_i = a_i)$ is called a *linear clause*. Sometimes, it is convenient to view a linear clause as a negation of a linear system, i.e., $C = \neg \bigwedge_{i=1}^{m} (f_i = a_i + 1)$. Note that a propositional clause $C = \bigvee_{i \in I} x_i \vee \bigvee_{j \in J} \overline{x_j}$ is a special case of a linear clause: we can write $C = \bigvee_{i \in I} (x_i = 1) \vee \bigvee_{j \in J} (x_j = 0)$.

Given a CNF formula F, a *Res($\oplus$) refutation of* F is a sequence of linear clauses $\{C_i\}_{i \in L}$ ending with the empty clause and where every clause either belongs to F or was derived from two previous clauses using one of the following two derivation rules: The *resolution rule*

$$\frac{C \vee (x = 1) \qquad D \vee (x = 0)}{C \vee D},$$

or the *semantic weakening rule*, which allows to derive from a linear clause C any linear clause D that semantically follows from it, i.e., such that any $x \in \{0,1\}^n$ satisfying C also satisfies D.

The *length* of a Res($\oplus$) refutation is the number L of linear clauses appearing in it. The *depth* d of a Res($\oplus$) refutation is the largest number of resolution steps among all the paths from axioms to the empty clause in the refutation. The proof system *depth-d Res($\oplus$)* is the subsystem of Res($\oplus$) consisting only of refutation of depth at most d, where d can depend on the number of variables n.

Shape-DAGs. We will also work with the top-down definition of the proof systems defined above, namely, *triangle-* and *affine-DAGs* solving *total search problems*. A *total search problem* is defined as a relation $S \subseteq \mathcal{I} \times \mathcal{O}$ over finite sets of inputs $\mathcal{I}$ and outputs $\mathcal{O}$, such that for every input $x \in \mathcal{I}$ there is an output $o \in \mathcal{O}$ with $(x, o) \in S$. Let $S^{-1}(o) = \{x \in \mathcal{I} \mid (x, o) \in S\}$.

For a given unsatisfiable CNF formula $F = C_1 \wedge \cdots \wedge C_m$, we consider the *falsified clause search problem* Search_F: Given an assignment $x \in \{0,1\}^n$ to the variables of F, output $i \in [m]$, such that the clause C_i is falsified under x. In detail, $(x, i) \in \mathrm{Search}_F$ if and only if $C_i(x) = 0$.

Definition 2.2 ([23]). Let $S \subseteq \mathcal{I} \times \mathcal{O}$ be a total search problem, $\mathcal{F} \subseteq 2^{\mathcal{I}}$. We call the elements of $\mathcal{F}$ *shapes*. A *shape-DAG* ($\mathcal{F}$-DAG) solving S is a rooted directed acyclic graph D of fan-out at most 2, where each node $v \in D$ is labeled by a shape $F_v \in \mathcal{F}$ such that

1. the root $r \in D$ is labeled by $F_r = \mathcal{I}$;
2. for a node v with children u, w, $F_v \subseteq F_u \cup F_w$;
3. and for every leaf ℓ, there is an output $o \in \mathcal{O}$ such that $F_\ell \subseteq S^{-1}(o)$.

The *depth* of D is the longest root-to-leaf path in D; its *length* is the number of nodes in it.

Given a bipartite domain $X \times Y$, a *triangle* $T \subseteq X \times Y$ is a set which can be written as $T = \{(x, y) \in X \times Y \mid a_T(x) \leq b_T(y)\}$, for some $a_T \colon X \to \mathbb{R}$, $b_T \colon Y \to \mathbb{R}$. A *triangle-DAG* is a shape-DAG where the shapes are all triangles. To consider triangle-DAGs for the falsified clause search problem, we split the variables of F into two parts and view $\{0,1\}^n$ as the corresponding product. Let $\mathrm{Search}_F^{X,Y} \subseteq (X \times Y) \times [m]$ denote $\{((x, y), i) \mid C_i(x, y) = 0\}$. As the following proposition states, lower bounds from cutting planes follow from lower bounds for triangle-DAGs.

Proposition 2.3 ([49, 30])**.** *A semantic cutting planes refutation of a given CNF formula F yields, for any partition of the variables $Vars(F) = X \sqcup Y$, a triangle-DAG solving $\mathrm{Search}_F^{X,Y}$ of the same length.*

An *affine subspace-DAG* is a shape-DAG where the shapes are affine subspaces of the domain $\mathbb{F}_2^n$. An *affine-DAG* is an affine subspace-DAG D, where every non-leaf node $v \in D$ has two outgoing edges, (v, w) and (v, w'), labeled $P_v = 0$ and $P_v = 1$, respectively. In addition, for the linear systems $\Psi_v, \Psi_w, \Psi_{w'}$ corresponding to the affine subspaces labeling v, w, w', it holds that Ψ_w is implied by $\Psi_v \wedge (P_v = 0)$ and $\Psi_{w'}$ is implied by $\Psi_v \wedge (P_v = 1)$. Similar to Proposition 2.3, lower bounds for affine-DAGs can be used to derive lower bounds for Res($\oplus$), as stated formally below.

Proposition 2.4 ([21])**.** *A Res($\oplus$) refutation of a given CNF formula F yields an affine-DAG solving Search_F of the same length and depth.*

2.3 Binary Encoding of Clique Formula

We encode the claim that a graph G has a clique of size k. We use the so-called *binary encoding* [41, 19], where for every $i \in [k]$, there are $\log|V|$ variables to point out the ith member of a purported k-clique. For a propositional variable x, we use the notation $(x \neq 0)$ to denote the literal x, and $(x \neq 1)$ to denote the literal $\overline{x}$.

Assume that the vertices v_i, v_j and v are represented as $v_{i,1}, \ldots, v_{i,\log|V|}$, $v_{j,1}, \ldots, v_{j,\log|V|}$, and $v_1, \ldots, v_{\log|V|}$ in binary, respectively. Let $x_{i,a}$, where $i \in [k]$ and $a \in [\log|V|]$, be propositional variables. The formula $\mathsf{BClique}(G, k)$ consists of the clauses

$$\bigvee_{a=1}^{\log|V|} (x_{i,a} \neq v_{i,a}) \vee \bigvee_{a=1}^{\log|V|} (x_{j,a} \neq v_{j,a}), \qquad \forall v_i \neq v_j \in V, i \neq j \in [k] : \{v_i, v_j\} \notin E, \tag{1}$$

$$\bigvee_{a=1}^{\log|V|} (x_{i,a} \neq v_a) \vee \bigvee_{a=1}^{\log|V|} (x_{j,a} \neq v_a), \qquad \forall v \in V, i \neq j \in [k]. \tag{2}$$

The clauses (1) are called *edge axioms*, and the clauses (2) are called *functionality axioms*. The edge axioms encode that two non-neighbors are not simultaneously chosen as clique members, while the functionality axioms ensure that all clique members are distinct.

Block encoding. We next consider a version of the binary clique formula, where cliques are required to have a "block-respecting" structure. That is, we partition the vertices of G into k *blocks* of equal size $V = V_1 \sqcup \cdots \sqcup V_k$. We then encode the claim that G has a k-clique with one vertex from every block. Such a clique is also called *transversal*.

Assume that v_i is represented as $v_{i,1}, \ldots, v_{i,\log n}$ and v_j is represented as $v_{j,1}, \ldots, v_{j,\log n}$. Let $x_{i,a}$, where $i \in [k]$ and $a \in [\log n]$, be propositional variables. Then the formula $\mathsf{BClique}_{\text{block}}(G,k)$ consists of the clauses

$$\bigvee_{a=1}^{\log n} (x_{i,a} \neq v_{i,a}) \vee \bigvee_{a=1}^{\log n} (x_{j,a} \neq v_{j,a}), \qquad \forall v_i \in V_i, v_j \in V_j : \{v_i, v_j\} \notin E. \tag{3}$$

The clauses (3) are edge axioms, and again, encode that for $(v_i, v_j) \notin E$, "v_i is not the clique member from block i or v_j is not the clique member from block j". Note that in the block encoding, we do not need functionality axioms, since the clique members are from distinct blocks by definition.

The following lemma shows that lower bounds for $\mathsf{BClique}_{\text{block}}(G,k)$ imply lower bounds for $\mathsf{BClique}(G,k)$. Therefore, for the rest of the paper, we will only consider the block encoding.

Lemma 2.5 ([3])**.** *Let $k \in \mathbb{N}^+$ and G be a graph. If there is a cutting planes (respectively, Res($\oplus$)) refutation of $\mathsf{BClique}(G,k)$ of length L and depth d, then there is a cutting planes (respectively, Res($\oplus$)) refutation of $\mathsf{BClique}_{block}(G,k)$ of length at most L and depth at most d.*

While the statement in [3] is for the unary encoding of clique formulas and the resolution proof system, its proof can be modified to obtain the lemma above. Indeed, one can obtain $\mathsf{BClique}_{\text{block}}(G,k)$ from $\mathsf{BClique}(G,k)$ by applying a random restriction, and both cutting planes and Res($\oplus$) are closed under random restrictions.

2.4 Communication Complexity

In this paper, we consider randomized (public coin) two party number-in-hand communication complexity. We quickly define the model and refer to [38, 44] for details.

Two players, *Alice* and *Bob*, are given inputs $x \in X$ and $y \in Y$, respectively. To solve a search problem $S \subseteq (X \times Y) \times \mathcal{O}$, they need to find $o \in \mathcal{O}$ such that $((x,y),o) \in S$, while communicating as few bits as possible. A *deterministic communication protocol* Π for S is a rooted binary tree as follows. Every internal node $v \in \Pi$ determines who is going to speak, and the spoken bit, which is a function of v and the input x or y (x if Alice speaks, y otherwise), determines which child of v the computation moves to. Every leaf $\ell \in \Pi$ is labeled by $o_\ell \in \mathcal{O}$, such that $((x,y),o_\ell) \in S$. We define a *randomized communication protocol* Π_r for S as a distribution over deterministic communication protocols, such that the computation is correct with constant probability. The *cost* of a deterministic protocol $|\Pi|$ is the maximum, over all inputs, of the number of bits communicated. The *cost* of a randomized protocol $|\Pi_r|$ is the maximum cost among the deterministic protocols in the distribution.

We study the communication complexity of $\text{Search}^{X,Y}_{\mathsf{BClique}(G,k)}$, where we split the variables of $\mathsf{BClique}(G,k)$ such that Alice and Bob each get half of the bits from every clique member, i.e., $X = Y = (\{0,1\}^{\log|V|/2})^k$, and denote this problem by $\mathsf{BClique}^{cc}(G,k)$. As in the proof complexity setting, a version of Lemma 2.5 applies and we may prove lower bounds for the block encoding, i.e., for $\mathsf{BClique}^{cc}_{\text{block}}(G,k)$.

Furthermore, we will make use of the characterization of randomized communication cost in terms of distributional communication complexity [52]: To prove a lower bound on the randomized communication cost, it is enough to prove a lower bounds on deterministic protocols with constant error probability for *any* input distribution. In particular, we will prove a lower bound for the uniform distribution.

3 Density Properties of Random Graphs

In the subsequent sections, we prove hardness results for graphs satisfying some combinatorial property related to the density of their neighbor sets. In this section, we define the properties we consider and prove that random graphs satisfy them.

Definition 3.1. Given $s, k, n > 0$, we say a graph G over k blocks of size n is *s-almost-complete* if for every $i, j \in [k]$, $i \neq j$, and $x_i, y_i, x_j \in \Sigma$, where $\Sigma = \{0,1\}^{\log(n)/2}$, there are at most s different y_j such that there is no edge between (x_i, y_i) and (x_j, y_j) in G.

We show that random dense graphs a.a.s. satisfy this property for an appropriate value of s.

Lemma 3.2. *For $p \in [0,1]$, if $\boldsymbol{G} \sim \mathcal{G}(n,p,k)$, then, a.a.s., $\boldsymbol{G}$ is $\max(2\sqrt{n}(1-p), 9e^2\log(kn))$-almost-complete.*

Proof. For fixed $i, j \in [k]$ and fixed x_i, y_i, and x_j, let $\boldsymbol{Z}$ be the random variable denoting number of y_j such that there is no edge between (x_i, y_i) and (x_j, y_j). First, suppose that $p \leq 1 - \frac{9\log(nk)}{\sqrt{n}}$. Chernoff's inequality implies that

$$\Pr[\boldsymbol{Z} \geq 2\sqrt{n}(1-p)] \leq \exp\left(-\frac{\sqrt{n}(1-p)}{3}\right).$$

Taking a union bound over every choice of i, j and of x_i, y_i and x_j, we find that the probability of a graph not satisfying the desired property is bounded by

$$\begin{aligned}\binom{k}{2}\sqrt{n}^3 \cdot \Pr[\boldsymbol{Z} \geq 2\sqrt{n}(1-p)] &\leq k^2 n^{3/2} \cdot \exp\left(-\frac{\sqrt{n}(1-p)}{3}\right) \\ &\leq k^2 n^{3/2} \cdot \exp(-3\log(nk)) \ll \frac{1}{nk}.\end{aligned}$$

Now, we consider the case where $p > 1 - \frac{9\log(nk)}{\sqrt{n}}$. Then a naive union bound gives

$$\begin{aligned}\binom{k}{2}\sqrt{n}^3 \cdot \Pr[\boldsymbol{Z} \geq 9e^2\log(kn)] &\leq \binom{k}{2}\sqrt{n}^3 \binom{\sqrt{n}}{9e^2\log(kn)}(1-p)^{9e^2\log(kn)} \\ &\leq \binom{k}{2}\sqrt{n}^3 \left(\frac{\sqrt{n}}{9e\log(kn)}\right)^{9e^2\log(kn)} \left(\frac{9\log(kn)}{\sqrt{n}}\right)^{9e^2\log(kn)} \\ &\leq k^2\sqrt{n}^3 \frac{1}{(kn)^{9e^2}}.\end{aligned}$$

This completes the proof of the lemma. □

The following property, which appeared in [20], is about the common neighborhood of small subsets of vertices. Similar definitions have also appeared in [3, 7, 2].

Definition 3.3. Given $\alpha, \beta, R, k, n > 0$, we say a graph G over k blocks of size n has (α, β, R)-*bounded* common neighborhoods if every $S \subseteq V(G)$ of size at most R and every block $i \in [k] \setminus B(S)$ satisfies

$$N^{\cap}(S, i) \in (1 \pm \beta)\alpha^{|S|}n.$$

For random graphs, this property expresses that the size of common neighborhoods of small sets behaves approximately as expected.

Lemma 3.4. *For $0 \leq \delta < 1$, $p \geq 1 - n^{-\delta}$, $\beta \geq \sqrt{\frac{18\log(kn)}{n^{1-\delta}}}$, if $\boldsymbol{G} \sim \mathcal{G}(n,p,k)$, then, a.a.s., $\boldsymbol{G}$ has (p,β,n^δ)-bounded common neighbors.*

Proof. For fixed $S \subseteq V(\boldsymbol{G})$ and $i \in [k] \setminus B(S)$, let $N^{\cap}_{\boldsymbol{G}}(S,i)$ be the random variable which is equal to the number of vertices of block i lying in the common neighborhood of S. By Chernoff's inequality,

$$\begin{aligned}\Pr_{\boldsymbol{G}}\left[\left||N^{\cap}_{\boldsymbol{G}}(S,i)| - p^{|S|}n\right| \geq \beta p^{|S|}n\right] &\leq 2 \cdot \exp\left(-\frac{\beta^2 p^{|S|}\cdot n}{3}\right)\\ &\leq 2 \cdot \exp\left(-\frac{\beta^2(1-|S|/n^\delta)\cdot n}{3}\right),\end{aligned}$$

where we use that $p^{|S|} \geq 1 - |S|n^{-\delta}$ for the second inequality. Then, by a union bound over all such sets S and blocks i, we conclude that $\boldsymbol{G}$ does not verify the property with probability at most

$$\begin{aligned}2k \cdot \sum_{j=1}^{n^\delta} \binom{kn}{j} \cdot \exp\left(-\frac{\beta^2(1-j/n^\delta)\cdot n}{3}\right) &\leq 2k \cdot \sum_{j=1}^{n^\delta} (kn)^j \cdot \exp\left(-\frac{\beta^2 n}{6}\right)\\ &\leq 4k \cdot (kn)^{n^\delta} \exp\left(-3n^\delta \log(kn)\right)\\ &\leq \exp\left(2n^\delta \log(kn) - 3n^\delta \log(kn)\right)\\ &\leq \exp(-n^\delta \log(kn)).\end{aligned}$$

□

4 Lower Bound for Cutting Planes

This section is devoted to proving our first result, stated formally below.

Theorem 4.1. *For any integers n and k, and for any real $p \in [0,1]$ if $\boldsymbol{G}$ is a graph sampled from $\mathcal{G}(n,p,k)$ then a.a.s. semantic cutting planes requires length*

$$\exp\left(\Omega\left(\min\left(\frac{1}{\sqrt{1-p}}, \frac{n^{1/4}}{\log(kn)}\right)\right)\right)$$

to refute $\mathsf{BClique}_{block}(G,k)$.

We quickly verify that Theorem 1.1 follows from the theorem stated above. For $p = 1 - n^{-1/3}$ and $k = n$, we have $N = nk = n^2$, and the bound in Theorem 4.1 becomes $2^{\Omega(N^{1/12})}$, and it applies to $\mathsf{BClique}(G,k)$ by Lemma 2.5. Also note that at this edge density, and with $k = n$, the formula $\mathsf{BClique}(G,k)$ is a.a.s. unsatisfiable, meaning the bound is non-trivial.

By Proposition 2.3 and Lemma 3.2, we obtain the above theorem by proving a size lower bound on triangle-DAG protocols solving the corresponding search problem, for the variable partition introduced in the preliminaries and over almost-complete graphs.

Theorem 4.2. *Let $s,k,n > 0$, $q = n^{1/4}/\sqrt{8s}$ and G over k blocks of size n and s-almost-complete then triangle-DAG protocols solving* $\mathrm{Search}^{X,Y}_{\mathsf{BClique}(G,k)}$ *require length* 2^{q-2}.

To see that Theorem 4.1 follows from Theorem 4.2 recall that, by Lemma 3.2, $\boldsymbol{G} \sim \mathcal{G}(n,p,k)$ is, a.a.s., s-almost-complete for $s = \max(2\sqrt{n}(1-p), 9e^2\log(kn))$. Applying Theorem 4.2 to such an s-almost-complete graph we have that

$$q = \frac{n^{1/4}}{\sqrt{8s}} = \Omega\left(\frac{n^{1/4}}{\sqrt{\max(\sqrt{n}(1-p), \log(nk))}}\right) = \Omega\left(\min\left(\frac{1}{\sqrt{1-p}}, \frac{n^{1/4}}{\log(kn)}\right)\right)$$

from which Theorem 4.1 follows. Before we present the proof of this theorem, let us establish some additional notation. Let G be a graph over k blocks each of size n with $V(G) = V_1 \sqcup \cdots \sqcup V_k$. Recall that for $X = Y = \Sigma^k$, where $\Sigma = \{0,1\}^{\log(n)/2}$, every $x = (x_i)_{1\le i\le k} \in X$, $y = (y_i)_{1\le i\le k} \in Y$, (x_i, y_i) uniquely identifies a vertex in the i-th block. Given $u \in V_i$ and $v \in V_j$, where $i \neq j$, we define $R_{u,v} \subseteq X \times Y$ to be the rectangle consisting of inputs $(x,y) \in X \times Y$ verifying $(x_i, y_i) = u$ and $(x_j, y_j) = v$, that is, $R_{u,v} \coloneqq \{x \in X \mid x_i = u_1 \dots u_{\log(n)/2} \wedge x_j = v_1 \dots v_{\log(n)/2}\} \times \{y \in Y \mid y_i = u_{\log(n)/2+1} \dots u_{\log(n)} \wedge y_j = v_{\log(n)/2+1} \dots v_{\log(n)}\}$. Let $\mathcal{R}$ be the set of all rectangles $R_{u,v}$ where u and v are non-adjacent. Additionally, for any $R \in \mathcal{R}$, we define $B(R)$ to be the set of vertices $\{u,v\}$ such that $R = R_{u,v}$. Note that $\mathcal{R}$ corresponds to the collection of sets of inputs corresponding to pre-images of solutions of $\mathrm{Search}^{X,Y}_{\mathsf{BClique}(G,k)}$.

For a triangle $T \subseteq X \times Y$ and an input $x \in X$, let T^x denote the *slice* $(\{x\} \times Y) \cap T$. We define the *block width* [50, 6] of x in T, denoted $\mathsf{bw}(T,x)$, to be the minimum number of unique blocks mentioned in a covering of T^x by rectangles from $\mathcal{R}$, i.e

$$\mathsf{bw}(T,x) = \min_{\substack{S \subseteq \mathcal{R} \\ T^x \subseteq \bigcup_{R\in S} S}} \left|\bigcup_{R\in S} B(R)\right|.$$

When T is evident from context, we simply call this the block width of x.

The proof of the main theorem is based on a bottleneck counting argument following a framework that appeared in previous work [28, 50, 6], adapted to our setting. We construct a partial function $\mu\colon X \sqcup Y \to V(\Pi)$, defined in Algorithm 1, for which we prove two properties. First, we show that a substantial fraction of all inputs must be assigned by μ to some node. We then prove that no single node can have many inputs assigned to it and conclude that there must be many nodes in the protocol.

Informally, for a threshold q, the map μ is constructed by traversing the nodes of the triangle-DAG in topological order from the leaves to the root and, whenever we encounter a $z \in X \sqcup Y$ such that $\mathsf{bw}(T,z) > q$, we let μ map z to that node in Π and remove z from all nodes not yet considered.

The first property of μ is straightforward to prove.

Lemma 4.3. *For $q \le n^{1/4}/\sqrt{s}$ and G s-almost-complete, then μ assigns at least $|X \sqcup Y|/4$ elements.*

Proof. Consider the rectangle $T'_r = X' \times Y'$ of unassigned inputs at the root. We prove that either X' is empty and therefore μ assigns $|X| = |X \sqcup Y|/2$ elements, or $|Y'| \le |Y|/2$ and thus μ assigns at least $|Y|/2 = |X \sqcup Y|/4$ elements.

If X' is empty, then we are done; otherwise, let x be any element in X'. By construction of μ, there exists a set $S \subseteq \mathcal{R}$ that covers $\{x\} \times Y'$ and mentions at most q blocks. Since the graph is s-almost-complete, and since every rectangle of S is indexed by a pair of vertices that do not share an edge,

$$|S| \le \binom{q}{2}\sqrt{n} \cdot s \le n/2\,, \tag{4}$$

Algorithm 1: Definition of μ

Input: input set $X \times Y$ and triangle-DAG Π
Output: partial map $\mu : X \sqcup Y \to V(\Pi)$

Let $X' := X$ and $Y' = Y$; /* the remaining coordinates not yet assigned by μ */
for *u in a topological ordering of $V(\Pi)$, starting from the sinks* **do**
  /* T_u is the triangle at vertex u in Π */
  $T'_u := T_u \cap (X' \times Y')$;
  for $x \in X'$ **do**
    if $\mathsf{bw}(T_u, x) > q$ **then**
      Let $\mu(x) := u$ and delete x from X';
      $T'_u := T_u \cap (X' \times Y')$;
  for $y \in Y'$ **do**
    if $\mathsf{bw}(T_u, y) > q$ **then**
      Let $\mu(y) := u$ and delete y from Y';
      $T'_u := T_u \cap (X' \times Y')$;
return μ

where we use that $q \le n^{\frac{1}{4}}/\sqrt{s}$. This implies that

$$\begin{aligned}\Pr_{\boldsymbol{y} \sim Y}\left[\boldsymbol{y} \in Y'\right] &\le \Pr_{\boldsymbol{y} \sim Y}\left[\exists R_{u,v} \in S \text{ s.t. } (x, \boldsymbol{y}) \in R_{u,v}\right] \\ &\le |S|/n \\ &\le 1/2\,,\end{aligned}$$

and therefore, $|Y'| < |Y|/2$, which concludes the proof. □

By definition of a triangle-DAG, the following claim is immediate from Algorithm 1.

Claim 4.4 ([50, 6])**.** *During the execution of Algorithm 1, for every u in the triangle-DAG Π and $z \in X' \cup Y'$, the block width of z in T'_u is at most $2q$.*

We now state the main lemma of this section.

Lemma 4.5. *Let $q \le n^{1/4}/\sqrt{8s}$ and G s-almost-complete. For all $u \in V(\Pi)$, μ maps at most $|X \sqcup Y| \cdot 2^{-q}$ elements of $X \sqcup Y$ to u.*

Assuming this lemma, Theorem 4.2 follows immediately.

Proof of Theorem 4.2. By Lemma 4.3, at least $|X \sqcup Y|/4$ elements from $|X \sqcup Y|$ are assigned to some vertex in Π by μ. On the other hand, Lemma 4.5 bounds the number of $z \in X \sqcup Y$ assigned to any one vertex in Π. Put together, we conclude that there must be at least $\frac{|X \sqcup Y|/4}{|X \sqcup Y| 2^{-q}} = 2^{q-2}$ vertices in Π. □

The rest of this section is dedicated to the proof of Lemma 4.5. Fix $u \in V(\Pi)$. Our goal is to show that the number of elements $x \in X$ assigned by μ to u is at most $2^{-q} \cdot |X|$. By symmetry, we have the same bound on the number of such $y \in Y$ and the lemma follows.

Algorithm 2: Definition of a tree $\mathcal{T}$ of potential coverings of T'^x

Input: Sets $X' \subseteq X$, $Y' \subseteq Y$, and triangle $T' \subseteq X' \times Y'$ s.t. $\mathsf{bw}(T', y) \leq 2q$ for all $y \in Y'$
Output: A tree $\mathcal{T}$ of coverings of T'^x for every $x \in X'$

```
Initialize 𝒯 as a tree with a single node labeled T';
Let Leaves := {T'};
while Leaves ≠ ∅ do
    Choose some T from Leaves and remove it from Leaves;
    Choose a y ∈ Y such that |T^y| is maximized;
    Let R_{u_1,v_1}, ..., R_{u_ℓ,v_ℓ} be a minimal covering for T^y with respect to block-width;
    /* note that |B({u_1, v_1, ..., u_ℓ, v_ℓ})| ≤ 2q                                  */
    for i ∈ [ℓ] do
        Let R := R_{u_i,v_i} = X_R × Y_R;
        Let T_R := T ∩ (X_R × (Y ∖ Y_R));
        Add an edge to 𝒯 labeled by (u_i, v_i) from the node of 𝒯 labeled T to a new leaf
         labeled by T_R;
        if T_R ≠ ∅ then
            Add T_R to Leaves;
return 𝒯
```

To obtain this bound, we run Algorithm 2 on $T' = (X' \times Y') \cap T_u$, with X' and Y' taking their values right before processing node u in Algorithm 1. From this, we obtain a tree $\mathcal{T}$ with the inner nodes labeled by triangles that are subsets of T' not covered yet, and with the edges labeled by rectangles in $\mathcal{R}$. It satisfies the following properties:

1. First, the set of edge label rectangles cover T', i.e., for every (x, y) in T', there is an edge labeled $R_{u,v}$ s.t. $(x, y) \in R_{u,v}$.

2. Second, if $(x, y) \in T'$ is in triangle T_1 at node t_1, and (x, y) is in triangle T_2 at node t_2, then either t_1 is a descendant of t_2 or vice-versa, and (x, y) is in every triangle labeling nodes between t_1 and t_2.

3. Furthermore, for every x, there is a unique path from the root to some leaf t in T' such that $\{R_{u,v} \mid R_{u,v} \text{ edge labels of path}\}$ covers T'^x.

For any node t of $\mathcal{T}$ whose path from the root is labeled by $R = (R_1, R_2, \ldots, R_\ell)$, the *block-depth* of t is the number of unique blocks mentioned by the rectangles in R. Then, the block-depth of an input $x \in X'$ is the maximal block-depth over nodes that x is consistent with.

Note that for every leaf t in $\mathcal{T}$, property 3 above implies that the block-depth of t is an upper bound on $bw(T'_u, x)$ for any $x \in X'$ consistent with t. To prove the lemma, we thus only have to bound the number of elements $x \in X'$ that are consistent with leaves of block-depth greater than q. Before doing so, we prove a claim that will allow us to simplify $\mathcal{T}$.

Claim 4.6. *Let t be a vertex in $V(\mathcal{T})$. If t has out-degree greater than one, then every child of t has block-depth greater than the block-depth of t.*

Proof. Let t' be a vertex that has the same block-depth as its parent t. The edge from t to t' must be labeled by a rectangle $R_{u,v}$ with vertices from blocks $i = B(u)$ and $j = B(v)$, both already mentioned on the path from the root to t. In particular, this means that all $x \in X'$ consistent with t agree on the ith and jth coordinate.

Now, let $y \in Y$ be as chosen when processing t in Algorithm 2. If there exists an $x' \in X'$ consistent with t such that $(x', y) \in R_{u,v}$, i.e., such that $(x'_i, y_i) = u$ and $(x'_j, y_j) = v$, this implies that for every $x \in X'$ consistent with t it holds that $(x_i, y_i) = u$ and $(x_j, y_j) = v$ since x agrees with x' in the ith and jth coordinate, and therefore $(x, y) \in R_{u,v}$. Hence T^y in Algorithm 2 is fully covered by $R_{u,v}$ and the out-degree of t is 1. □

We can now modify $\mathcal{T}$ by removing every vertex in $\mathcal{T}$ with a child of the same block-depth and replacing it with that child. This leaves us with a new tree $\mathcal{T}'$ where every path from the root to a leaf has strictly increasing block-depth. Note that the block-depth of an input $x \in X'$ is the same in $\mathcal{T}'$ as in $\mathcal{T}$. In order to conclude the proof, it is now sufficient to prove the following claim.

Claim 4.7. *The number of vertices at block-depth d in $\mathcal{T}'$ is at most $\left(\frac{\sqrt{n}}{2}\right)^d$.*

Proof. We prove this by induction over the block-depth d. For $d \leq 1$, the claim follows since the only vertex at block-depth 0 is the root of $\mathcal{T}'$, and there are no vertices at block-depth 1.

For the induction step, note that the children of a vertex $t \in V(\mathcal{T}')$ at block-depth d' either have block-depth $d' + 1$ or $d' + 2$. We will argue that any vertex $t \in V(\mathcal{T}')$ at block-depth d' can have at most $\sqrt{n}/4$ children of block-depth $d' + 1$ and at most $n/4$ children of block-depth $d' + 2$. This would imply the claim since we can conclude that, for $d \geq 2$, there are at most

$$\frac{\sqrt{n}}{4}\left(\frac{\sqrt{n}}{2}\right)^{d-1} + \frac{n}{4}\left(\frac{\sqrt{n}}{2}\right)^{d-2} \leq \left(\frac{\sqrt{n}}{2}\right)^{d}$$

vertices at block-depth d, where we apply the induction hypothesis to bound the number of vertices at level $d - 1$ and $d - 2$.

Let $t \in V(\mathcal{T}')$ be a vertex at block-depth d'. Consider the $y \in Y$ chosen when processing this vertex in Algorithm 2 and the rectangles $R_{u_1,v_1}, \ldots, R_{u_d,v_d}$ chosen to be a minimal covering for T^y. These rectangles are the labels of the out-edges of t. Since the block-width of y is at most $2q$, there is a set of indices $S \subseteq [k]$, with $|S| \leq 2q$, such that $B(\{u_1, v_1, \ldots, u_d, v_d\}) \subseteq S$.

We first argue that t has at most $\sqrt{n}/4$ children of block-depth $d' + 1$. A child of t which is incident to an edge labeled $R_{u,v}$ is at level $d' + 1$ if and only if exactly one of $B(u)$ or $B(v)$ is a block that already appeared in the path from the root of $\mathcal{T}$ to t. For each pair of indices $i, j \in S$, where i already appears in the path from the root to t and j does not, we bound the number of rectangles $R_{u,v}$ such that $\{i, j\} = B(\{u, v\})$. Since i appears earlier in the path, all $x \in X'$ consistent with t agree on the ith coordinate and thus x_i is fixed for all rectangles $R_{u,v}$ labeling the out-edges of t with $B(\{u, v\}) = \{i, j\}$. Since G is s-almost-complete, and both y_i and y_j are clearly also fixed in T^y, then there are at most s such rectangles. Considering every possible pair of indices (i, j), of which there are at most $\binom{|S|}{2} \leq 2q^2$, we conclude that t has at most $2q^2 s \leq \frac{\sqrt{n}}{4}$ children of block-depth $d' + 1$.

It remains to argue that t has at most $n/4$ children of block-depth $d' + 2$. A child of t which is adjacent to an edge labeled $R_{u,v}$ is at level $d' + 2$ if and only if neither $B(u)$ nor $B(v)$ is a block that appears in the path from the root of $\mathcal{T}$ to t. For each pair of indices $i, j \in S$, where neither i nor j appears in the path from the root to t, and for each $x_i \in [\sqrt{n}]$ we use that G is s-almost-complete

(again using that y_i and y_j are fixed in T^y) to obtain that the number of rectangles $R_{u,v}$ labeling out-edges of t such that $B(u) = i$, $B(v) = j$ and $(x_i, y_i) = u$ is at most s. Considering every possible pair of indices (i, j) and every $x_i \in [\sqrt{n}]$, we conclude that t has at most $2q^2\sqrt{n}s \leq n/4$ children of block-depth $d' + 2$. □

Given this claim we can conclude that Lemma 4.5 holds. Indeed, every $x \in X'$ mapped by μ must be consistent with a leaf in $\mathcal{T}$ of block-depth larger that q, and since block-depth can only increase by 1 or 2 from parent to child in $\mathcal{T}$, every $x \in X'$ consistent with a leaf of block-depth greater that q is consistent with a vertex of block-depth exactly $q + 1$ or $q + 2$. Furthermore, only $|X|/\sqrt{n}^{\ell}$ inputs $x \in X'$ can be consistent with a vertex $t \in V(\mathcal{T}')$ at block-depth ℓ since all $x \in X'$ consistent with t coincide on the ℓ blocks mentioned on the path to t. From Claim 4.7, it follows that at most

$$\left(\frac{\sqrt{n}}{2}\right)^{q+1} \cdot \frac{|X|}{(\sqrt{n})^{q+1}} + \left(\frac{\sqrt{n}}{2}\right)^{q+2} \cdot \frac{|X|}{(\sqrt{n})^{q+2}} \leq \cdot 2^{-q}|X|$$

elements of $|X|$ are mapped to u by μ. By symmetry the same holds for $|Y|$ and the lemma follows.

5 Lower Bound for Bounded Depth Resolution over Parities

In this section, we prove our second result, stated formally below.

Theorem 5.1. *Let $\gamma > 0$ be a constant, let $p \in [1 - n^{-\gamma}, 1]$ and $k \in \mathbb{N}$ be functions of n, where $k < \exp(n^{o(1)})$ and let $\boldsymbol{G} \sim \mathcal{G}(n, p, k)$. Then asymptotically almost surely, for any Res($\oplus$)-refutation of depth D and size S that refutes $\mathsf{BClique}(G, k)$ it must hold that*

$$D\sqrt{\log S} \geq \begin{cases} \Omega(n^{3\gamma/2}), & \text{if } \gamma < 1/2 \\ \Omega(\frac{n^{\frac{1+\gamma}{2}}}{\sqrt{\log kn}}), & \text{if } 1/2 \leq \gamma < 1 \\ \Omega(n^{1-\varepsilon}), & \text{if } \gamma \geq 1 \end{cases}$$

for any constant $\varepsilon > 0$.

We start by verifying that Theorem 1.2 from the introduction follows from this theorem. Let $\gamma = 1/3$ and $k = n^{1/3+\varepsilon/100}$ for $\varepsilon > 0$ given in Theorem 1.2. Then $N = kn = n^{4/3+\varepsilon/100}$, and $\mathsf{BClique}(G, k)$ has $m = k \log N = O(n^{1/3+\varepsilon/100} \log n)$. So, if

$$D \leq m^{1.5-\varepsilon} = O(n^{(1/3+\varepsilon/100)(1.5-\varepsilon)} \log^{1.5-\varepsilon} n) = o(n^{3\gamma/2-\varepsilon/4}),$$

we have that $\log S = \Omega(\text{poly}(n)) = \Omega(\text{poly}(N))$. Therefore, sampling $\mathbf{G}$ using $p = 1 - n^{-1/3}$ and $k = n^{1/3+\varepsilon/100}$ a.a.s. yields an unsatisfiable formula requiring size $S = 2^{N^{\Omega(1)}}$ to refute in depth at most $m^{1.5-\varepsilon}$, as required.

To prove Theorem 5.1 we prove the slightly more general statement below.

Theorem 5.2. *Let $\alpha > 0$, $0 < \beta < 1/300$ be real numbers and let $R \leq 1/100(1 - \alpha)$ be a positive integer. Given a graph G with (α, β, R)-bounded common neighborhoods and a Res($\oplus$)-refutation of $\mathsf{BClique}(G, k)$ of depth D and size S, it must hold that*

$$D\sqrt{\log S} \geq \Omega(R \cdot \min(\sqrt{R}, 1/\beta))$$

The remainder of this section is dedicated to the proof of this theorem. But first we show that Theorem 5.1 follows from Theorem 5.2.

Proof of Theorem 5.1. Let $\alpha = p \geq 1 - n^{-\gamma}$. We choose δ differently according to γ. Let $\delta = \gamma$ if $\gamma < 1$, otherwise, let $\delta = 1 - \varepsilon$. Now let $R = n^{\delta}/100$ and let $\beta = \sqrt{\frac{18\log(kn)}{n^{1-\delta}}}$. By Lemma 3.4, $\boldsymbol{G}$ will asymptotically almost surely have (α, β, R)-bounded common neighbourhoods. Note that $R \leq 1/100(1-\alpha)$ and so we can apply Theorem 5.2 and obtain the lower bound

$$D\sqrt{\log S} \geq \Omega(R \cdot \min(\sqrt{R}, 1/\beta)) .$$

We split the analysis into cases depending on γ. If $\gamma < 1/2$, note that $1/\beta \geq n^{-\delta/2} = n^{-\gamma/2} \geq \sqrt{R}$ for large enough n and $k < \exp(n^{o(1)})$, and therefore we get the first lower bound, that is, $D\sqrt{\log S} \geq \Omega(n^{\gamma} \cdot n^{\gamma/2}) = \Omega(n^{3\gamma/2})$.

If $\gamma \geq 1/2$, note that $1/\beta \leq n^{(1-\delta)/2} \leq n^{\delta/2} = 10\sqrt{R}$ for large enough n and using the assumption $k < \exp(n^{o(1)})$. This implies that we get the bound

$$D\sqrt{\log S} \geq \Omega\left(n^{\gamma} \cdot \frac{n^{\frac{1-\gamma}{2}}}{\sqrt{\log nk}}\right) = \Omega\left(\frac{n^{\frac{1+\gamma}{2}}}{\sqrt{\log nk}}\right)$$

for $1/2 \leq \gamma < 1$ and the bound

$$D\sqrt{\log S} \geq \Omega\left(n^{\delta} \cdot \frac{n^{\frac{1-\delta}{2}}}{\sqrt{\log nk}}\right) = \Omega\left(\frac{n^{\frac{1+\delta}{2}}}{\sqrt{\log nk}}\right) = \Omega\left(\frac{n^{1-\varepsilon/2}}{\sqrt{\log nk}}\right) \geq \Omega\left(n^{1-\varepsilon}\right)$$

for $\gamma > 1$, where for the last inequality we assume n large enough and $k < \exp(n^{o(1)})$.

□

The proof of Theorem 5.2 adapts the proof by Byramji and Impagliazzo [12] of a Res($\oplus$) size-depth tradeoff for the BPHP. They do this by showing that in a small Res($\oplus$) refutation of BPHP, it is possible to repeatedly perform long random walks without reaching a sink.

5.1 Technical Lemmas and Definitions

Here we collect some technical preliminaries that are used only in this section. We start by recalling the notion of safe systems, introduced by [21].

We say that a set of linear forms F over variables $x_{i,j} \in (\{0,1\}^{\log n})^{[k]}$ is *dangerous* if the number of distinct blocks from which variables appear in F is smaller than the size of F. If no subset of F is dangerous we say that F is *safe*. It turns out that safe sets have a very nice characterization making them nice to work with.

Lemma 5.3 ([21])**.** *Let L be a collection of k independent linear forms and M the corresponding coefficient matrix. V is safe if and only if we can pick k variables, no two from the same block, such that the corresponding columns in M are linearly independent.*

To handle non-safe systems we use the notion of closure. The *closure* of F, $\mathrm{Cl}(F) \subseteq [k]$ is defined as the minimal set (with respect to inclusion) of blocks such that $F[\setminus \mathrm{Cl}(F)]$ is safe, where $F[\setminus S]$ denotes the set of linear forms F after setting variables belonging to a block in S to zero. It turns out that the closure is uniquely defined [21, Lemma 4.1]. In an abuse of notation we will, for a linear system Ψ, write $\mathrm{Cl}(\Psi)$ to refer to the closure of the set of linear forms F defining Ψ.

Lemma 5.4 ([21]). *If F is a collection of linear forms, then* $|\mathrm{Cl}(F)|+\dim(\langle F[\setminus \mathrm{Cl}(F)]\rangle) \leq \dim(F)$.

To assign variables in a way that satisfies linear systems, we need the concept of affine restrictions. A (block respecting) *affine restriction* ρ is a partial assignment that for some set $F \subseteq [k]$ sets all variables $x_{i,j}$, $i \in [k] \setminus F$ as an affine function of the unassigned variables $x_{i,j}$, $i \in F$. In this paper all affine restrictions will be block respecting and will for brevity just be called affine restrictions. For a linear system Ψ we use $\Psi|_\rho$ to denote the system after substituting variables according to ρ.

We introduce the search problem $\mathrm{NonEdge}_M^G \subseteq (\{0,1\}^{\log n})^{[k]\setminus B(M)} \times \binom{V(G)}{2}$, where M is some set of vertices of G such that vertices in M from distinct blocks in G always have edges between them. As input we are given an assignment to the blocks outside of the support of M such that every selected vertex lies in $N^\cap(M, [k] \setminus B(M))$. We are then asked to output two selected vertices not connected by an edge. When the graph G is clear from context, we simply write $\mathrm{NonEdge}_M$. Observe that $\mathrm{NonEdge}_M^G$ is equivalent to $\mathrm{Search}_{\mathsf{BClique}(G',k-|B(M)|)}$ where G' is the subgraph of G induced by $N^\cap(M, [k] \setminus B(M))$. Thus, any affine-DAG solving the falsified clause search problem also solves $\mathrm{NonEdge}_\emptyset^G$, as such it is enough to find lower bound for this problem. The following lemma was stated in [12] with $A_1 = A_2 = \cdots = A_n$. Their proof still works for our slightly more general formulation.

Lemma 5.5 ([12, Lemma 4.6]). *Let Ψ be a linear system on $(\{0,1\}^{\log n})^k$ whose rank is r. For each $i \in [k]$ let $A_i \subseteq \{0,1\}^{\log n}$ be such that $|A_i| \geq 2n/3$. Let $\boldsymbol{x}_i \sim A_i$ independently for each $i \in [k]$ Then*

$$\Pr[\boldsymbol{x} \text{ satisfies } \Psi] \leq \left(\frac{3}{4}\right)^r.$$

Lastly, we will later need to use a Chernoff bound on the sum of dependent variables. For this the following lemma will be useful.

Lemma 5.6 ([42, Lemma 17.3]). *Let $\boldsymbol{X}_1, \boldsymbol{X}_2 \ldots, \boldsymbol{X}_n$ be a sequence of random variables in an arbitrary domain and let $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_n$ be a sequence of binary variables with the property that $\boldsymbol{Y}_i = \boldsymbol{Y}_i(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_i)$. If*

$$\Pr[\boldsymbol{Y}_i = 1 \mid \boldsymbol{X}_1, \ldots, \boldsymbol{X}_{i-1}] \leq p,$$

then

$$\Pr\left[\sum_{i=1}^n \boldsymbol{Y}_i > k\right] \leq \Pr\left[\mathrm{Ber}^n(p) > k\right].$$

5.2 Proof of Theorem 5.2

Throughout this section we let $\alpha, \beta < 1/300$ be positive real values and $R < 1/100(1-\alpha)$ a positive integer. Furthermore, we assume G is a fixed graph with (α, β, R)-bounded common neighborhoods. Note that this choice of parameters ensures that for every vertex set $|M| \leq R$, $|N^\cap(M, i)| \leq 2n/3$.

The proof of Theorem 5.2 follows the same strategy as [12]. Given an affine-DAG solving $\mathrm{NonEdge}_M$ we wish to find an affine-DAG of much smaller depth solving a slightly smaller instance $\mathrm{NonEdge}_{M^*}$; repeating this process shows that the original depth must have been large. Formally, we use the following lemma which we will spend the remainder of the section proving.

Lemma 5.7. *Let $M \subseteq V(G)$ be a set of vertices and suppose there is an affine-DAG Π solving NonEdge_M whose depth is at most D and size is at most S. Then $D > (R - |M|)/8$.*

Moreover, for some universal constant C it holds that if $|M| + 8d \leq R$ for $d = \lfloor \sqrt{\log S} \cdot \min(\sqrt{R}, 1/\beta) \rfloor$ then there exists a set $M^* \supseteq M$ of size $|M^*| \leq |M| + C \log S$ and an affine-DAG Π' solving NonEdge_{M^*} whose depth is at most $D - d$ and size is at most S.

First we show how Theorem 5.2 follows from Lemma 5.7.

Proof of Theorem 5.2. A Res($\oplus$)-refutation of $\mathsf{BClique}(G, k)$ of depth D and size S immediately gives an affine Π solving $\text{NonEdge}_\emptyset$. Let $d = \lfloor \sqrt{\log S} \cdot \min((1-\alpha)^{-1/2}, 1/\beta) \rfloor$ we start by considering the case when $8d \geq R/2$. Then

$$R \leq 16d \leq \sqrt{\log S} \min(\sqrt{R}, 1/\beta) \leq \sqrt{\log S}\sqrt{R}.$$

This means $\sqrt{\log S} \geq \sqrt{R}$ and the desired bound follows immediately since by Lemma 5.7 we have that $D > R/8$.

Now consider the case $8d < R/2$. We can here use Lemma 5.7 to find a set M_1 of size $|M_1| = C \log S$ and an affine-DAG Π_1 of depth at most $D - d$ and size at most S solving NonEdge_{M_1}. We keep applying Lemma 5.7 iteratively in this way, using $M = M_{t-1}$ to find $M_t = M^*$ of size $|M_t| \leq tC \log S$ and Π_t of depth at most $D - dt$ and size at most S. We repeat this process as long as the condition of Lemma 5.7 holds, i.e., as long as $|M_t| + 8d \leq R$. Thus the final t must satisfy $tC \log S + 8d \geq R$ which means $t \geq R/2C \log S$. But at that point we have an affine-DAG Π_t of depth $D - dR/2C \log S$ solving M_t. Since an affine-DAG cannot have negative depth we find that $D \geq \Omega(dR/\log S) = \Omega(R \cdot \min(\sqrt{R}, 1/\beta)/\sqrt{\log S})$ and thus the theorem follows. □

Now all that remains is to prove Lemma 5.7. We do so by constructing a random walk according to Algorithm 3, over our affine-DAG. We require this random walk to satisfy several properties. During the walk, we build an affine restriction, which sets relatively few variables and satisfies all linear systems labeling nodes on the walked path. We will ensure that the walk visits all nodes with the same probability, as answering according to a uniformly sampled fixed input. We show that, with a significant probability, the walk does not witness a missing edge. On the other hand, if the affine-DAG is not sufficiently large, some node v must be visited with a significant probability in one of these successful runs. This is only possible if the rank of the system Ψ labeling v is not too large.

We can then use the fact that a small rank Ψ can be satisfied by restricting a few blocks. We combine this with the fact that v was reached in a successful run to find a valid instance of NonEdge_{M^*} that is solved by the dag rooted at v, i.e., a DAG of lower depth as desired.

We start by presenting Algorithm 3. This is essentially the same algorithm as [12, Algorithm 1] with two main differences. First, we require x_i to lie in $N^\cap(M, i)$ rather than an arbitrary set A. Second, we say that the algorithm *fails* if it assigns blocks in a way that could select vertices not connected by an edge rather than if it finds a collision between pigeons.

Let us state two properties of Algorithm 3 that follow immediately from the analysis of [12], the difference lying in the fact that we use $N^\cap(M, i)$ as a set of valid assignments to the x_i rather than an arbitrary set A.

Lemma 5.8 ([12, Lemma 4.1])**.** *At the end of every iteration of the while loop in Algorithm 3*

1. *the equations in L uniquely determine $x_{i,j}$ for all $i \in [k] \setminus F$, $j \in [\log n]$ as an affine function of $x_{i,j}$, $i \in F$, $j \in [\log n]$;*

Algorithm 3: Random walk simulator

Input: M: Set of vertices. T: Parity decision tree.

$C \leftarrow []$; // List of vertices used during simulation
$F \leftarrow [k] \setminus B(M)$; // Unassigned blocks
$L \leftarrow \emptyset$; // Collection of equations
$v \leftarrow$ root of T;
while *v is not a leaf* **do**
  $P' \leftarrow$ query at v;
  $P \leftarrow P'$ after substituting according to L;
  if *P is a constant, $b \in \{0,1\}$* **then**
    update v according to b;
  else
    $(i,j) \leftarrow \min\{(i,j) \in F \times [\log n] \mid x_{i,j} \text{ appears in } P\}$;
    pick y uniformly from $N^{\cap}(M,i)$;
    $L \leftarrow L \cup \{x_{i,h} = y_h \mid h \in [\log n] \setminus \{j\}\}$;
    if $y^{\oplus j} \notin N^{\cap}(M,i)$ **then**
      $L \leftarrow L \cup \{x_{i,j} = y_j\}$;
      append $\{(i,y)\}$ to C;
    else
      pick $b \in \{0,1\}$ uniformly at random;
      $L \leftarrow L \cup \{P = b\}$;
      append $\{(i,y),(i,y^{\oplus j})\}$ to C;
      update v according to b;
    $F \leftarrow F \setminus \{i\}$
if *there exists $j_1 \neq j_2$ such that $(i_1,y_1) \in C_{j_1}, (i_2,y_2) \in C_{j_2}$ and there is no edge between vertex y_1 in block i_1 and y_2 in block i_2* **then**
  return FAIL;
else
  return v, C, L, F;

2. *fixing the variables in $x_{i,j}$, $i \in F$, $j \in [\log n]$ and setting the remaining variables as determined by L ensures that $x_{i'} \in N^{\cap}(M,i')$ for all $i' \in [k] \setminus F$;*

3. *L implies all parity constraints on the path from the root to the current node.*

Lemma 5.9 ([12, Lemma 4.2])**.** *Let W_v be the event that node v of T is visited by Algorithm 3, Let $\boldsymbol{x}_i \sim N^{\cap}(M,i)$ independently for each $i \in [k] \setminus B(M)$, and V_v be the event that running T on $\boldsymbol{x}$ reaches v. Then for every $v \in T$, we have $\Pr[W_v] = \Pr[V_v]$.*

Now, we will show that the algorithm does not fail with a non-negligible probability (i.e., the walk does not witness a missing edge).

Lemma 5.10. *Let T be a depth-d parity decision tree and M a set of vertices. If $|M| + 8d \leq R$, then the probability that Algorithm 3 does not return FAIL when run on T and M is at least $\exp(-\Omega(d\beta + d^2(1-\alpha))$.*

Proof. To bound the probability of failure we consider two cases depending on the number of iterations of the while loop on Line 5. First consider the case where the algorithm runs for more than $4d$ iterations. We wish to show that the probability of this happening is small. To see this, notice that in every iteration we update v except if we reach Line 14 with $y^{\oplus j} \notin N^{\cap}(M, i)$. But since $|M| \leq R \leq 1/100(1-\alpha)$, it follows by the bounded common neighborhood property of the underlying graph that $|N^{\cap}(M, i)| > 2n/3$. Since y is picked uniformly from $N^{\cap}(M, i)$, the probability of $y^{\oplus j} \notin N^{\cap}(M, i)$ is bounded above by $1/2$. Let $\boldsymbol{X}_t$ be the indicator of the event that we update v on iteration t. Then, since $\mathbb{E}[\boldsymbol{X}_t \mid \boldsymbol{X}_{t-1}, \ldots, \boldsymbol{X}_1] \geq 1/2$ we can apply Lemma 5.6 with $\boldsymbol{Y}_i = \boldsymbol{X}_i$ to find $\Pr[\sum_{t=1}^{4d} \boldsymbol{X}_t < d] \leq \Pr[\mathrm{Ber}^{4d}(1/2) < d] \leq \exp(-d/4)$.

We now wish to bound the probability of finding a non-edge in the first $4d$ iterations. We start by bounding the probability in the tth iteration. Let i and j be the indices selected on Line 11. In every iteration, we append a set of at most size 2 to C, and so by the tth iteration the union of all sets in C contains at most $2(t-1)$ elements. Call this union C_t. By the bounded common neighborhood property, we know that the number of vertices in the ith block that lie in the common neighborhood of vertices in C_t and M is at least $(1-\beta)\alpha^{|M|+2(t-1)}$ while the size of $N^{\cap}(M, i)$ is at most $(1+\beta)\alpha^{|M|}$. Thus, the probability of y having an edge to a vertex in C_t is at least $\frac{1-\beta}{1+\beta}\alpha^{2(t-1)} \geq (1-2\beta)\alpha^{2(t-1)}$. Similarly, the probability of $y^{\oplus j}$ either not lying in $N^{\cap}(M, i)$ or having an edge to some vertex in C_t is also bounded below by this. By a union bound, the probability of not failing at the tth iteration is at least $2(1-2\beta)\alpha^{2(t-1)} - 1$. Note that, for $t \leq 4d$,

$$\alpha^{2(t-1)} \geq \alpha^{8d} \geq \alpha^{R} \geq \alpha^{1/2(1-\alpha)} \geq \alpha^{-1/2\log\alpha} = 1/\sqrt{2}.$$

Hence, using the fact that $\beta < 1/20$, we have that

$$2(1-2\beta)\alpha^{2(t-1)} - 1 \geq \sqrt{2}(1-1/10) - 1 \geq 1/4\,.$$

The probability of not failing after $4d$ iterations is thus at least

$$\begin{aligned}
\prod_{t=1}^{4d}\left(2(1-2\beta)\alpha^{2(t-1)} - 1\right) &\geq \prod_{t=1}^{4d}\exp(4((1-2\beta)\alpha^{2t} - 1)) \\
&\geq \exp(4\sum_{t=1}^{4d}((1-2\beta)\alpha^{2t} - 1)) \\
&\geq \exp(4\sum_{t=1}^{4d}((1-2\beta)(1-2t(1-\alpha)) - 1)) \\
&\geq \exp(-8\sum_{t=1}^{4d}(\beta + t(1-\alpha))) \\
&\geq \exp(-32d\beta - 64d^2(1-\alpha))\,,
\end{aligned}$$

where for the first inequality, we use that $1 - x \geq \exp(-2x)$ for $x < 0.75$. In total, the probability of both not failing within the first $4d$ iterations and terminating within $4d$ steps is bounded from below by $\exp(-32d\beta - 64d^2(1-\alpha)) - \exp(-d/4) \geq \Omega(\exp(-32d\beta - 64d^2(1-\alpha)))$, using the fact that $\beta \leq 1/300$ and $8d \leq R \leq 1/100(1-\alpha)$. □

We now wish to show that it is possible to use a successful run of Algorithm 3 to find an affine restriction satisfying the reached node.

Lemma 5.11. *Let v, C, L, F be returned by a successful run of Algorithm 3 when run on T and M. Let Ψ be the linear system labeling the node in the affine-DAG corresponding to v, and let r be the rank of Ψ. If $|M| + 2r \leq R$, then there exists a set $F' \subseteq [k]$ and an affine restriction ρ fixing blocks $[k] \setminus F'$ satisfying the following conditions:*

1. *The number of blocks fixed by ρ, say s, is at most r.*
2. *Ψ is implied by Ψ_ρ (the linear system equivalent to the affine restriction ρ).*
3. *There exists a set $M' \subseteq V(G) \setminus M$ such that $|M'| \leq 2s$ and for any assignment to F', if we set the blocks $[k] \setminus F'$ according to ρ, all the vertices assigned to $[k] \setminus F'$ lie in M' and form a complete subgraph of G.*

Proof. We start by fixing the variables in $\mathrm{Cl}(\Psi) \cap F$. Let $\widehat{M}$ be the set of vertices assigned by the simulation to the blocks in $\mathrm{Cl}(\Psi) \setminus F$, i.e., the yth vertex of the ith block for every $(i, y) \in C_t$ for some t and $i \in \mathrm{Cl}(\Psi) \setminus F$. Note that $|\widehat{M}| \leq 2|\mathrm{Cl}(\Psi) \setminus F|$. We now assign x_i, for every $i \in \mathrm{Cl}(\Psi)$, such that $x_i \in N^\cap(M \cup \widehat{M}, i)$ and all vertices selected form a complete subgraph. Since G has (α, β, R)-bounded common neighborhoods, this can be done as long as $|M \cup \widehat{M}| + |\mathrm{Cl}(\Psi) \cap F| \leq R$. But this holds since

$$|M \cup \widehat{M}| + |\mathrm{Cl}(\Psi) \cap F| \leq |M| + 2|\mathrm{Cl}(\Psi) \setminus F| + |\mathrm{Cl}(\Psi) \cap F| \leq |M| + 2|\mathrm{Cl}(\Psi)| \leq |M| + 2r \leq R$$

Set all blocks in $F \setminus \mathrm{Cl}(\Psi)$ arbitrarily and extend this to a full assignment x according to L. Since all blocks in F have been set, this extension exists and is unique by Lemma 5.8. Furthermore, L implies all parity constraints on some path to node v which in particular means that L implies Ψ. Thus, x satisfies Ψ. Let ρ_1 be the partial assignment setting every block in $\mathrm{Cl}(\Psi)$ according to x. By construction this assignment will not falsify any equation in Ψ. For the second step, we consider $\Psi' = \Psi|_{\rho_1}$. Since $\mathrm{Cl}(\Psi)$ is fixed by ρ_1, Ψ' is safe. By Lemma 5.3, there exists a set of variables of size $\dim(\Psi)$ belonging to different blocks and whose corresponding columns in Ψ are linearly independent. Let X be this set of variables and B the corresponding set of blocks. We now wish to fix all but one variable of every block in B, leaving the variables in X untouched, such that no matter what the variables of X are set to, the vertices in the blocks B all have edges to each other and to all vertices in M and the ones fixed by ρ_1. We do this one block at a time. When we get to block i, we fix $\log n - 1$ variables, such that both vertices $\{v_i, v_i'\}$, we could potentially select when setting the last variable, lie in the common neighborhood of M, of the vertices selected by ρ_1 and of all previous $\{v_{i'}, v_{i'}'\}$. By the pigeonhole principle, we can find such an assignment to the $\log n - 1$ variables of block i, as long as the common neighborhood of all potential previous selections at block i is larger than $n/2$. But the number of previously selected vertices is at most $|M| + |\mathrm{Cl}(\Psi)| + 2(\dim(\Psi') - 1)$ which by Lemma 5.4 can be bounded by

$$|M| + |\mathrm{Cl}(\Psi)| + 2(\dim(\Psi') - 1) \leq |M| + 2(|\mathrm{Cl}(\Psi)| + \dim(\Psi') - 1) \leq |M| + 2r < R.$$

This means that the common neighborhood will have size greater than $n/2$ as desired.

Call this partial assignment of $(\log n - 1)|B|$ variables ρ_2. To find the final ρ we use the fact that Ψ' is safe to solve Ψ expressing the variables in X as an affine function of the variables outside of B. We now let ρ be this affine restriction combined with ρ_1 and ρ_2. □

With this we have both shown that the probability of succeeding is significant and that when we succeed, we can extract an affine restriction satisfying the system labeling the reached node. We can now combine these facts to prove Lemma 5.7, restated here for convenience.

Lemma 5.7 (Restated). *Let $M \subseteq V(G)$ be a set of vertices and suppose there is an affine-DAG Π solving NonEdge_M whose depth is at most D and size is at most S. Then $D > (R - |M|)/8$.*

Moreover, for some universal constant C it holds that if $|M| + 8d \leq R$ for $d = \lfloor \sqrt{\log S} \cdot \min(\sqrt{R}, 1/\beta) \rfloor$ then there exists a set $M^* \supseteq M$ of size $|M^*| \leq |M| + C \log S$ and an affine-DAG Π' solving NonEdge_{M^*} whose depth is at most $D - d$ and size is at most S.

Proof of Lemma 5.7. We start by showing that $D > (R - |M|)/8$. Assume for contradiction that $D \leq (R - |M|)/8$ and expand Π into a parity decision tree T of depth D solving NonEdge_M. Since $|M| + 8D \leq R$, Lemma 5.10 gives a non-zero probability of Algorithm 3 succeeding. But after a successful run returning (v, C, L, F), an input x that reaches v can be constructed by setting the variables in F arbitrarily and setting the remaining variables according to L. Since the variables in blocks of F can be set arbitrarily, v cannot be labeled with a valid answer to NonEdge_M mentioning vertices in F. But the vertices assigned outside of F are all connected. This contradicts the assumption that T solves NonEdge_M and $D > (R - |M|)/8$ must hold.

To prove the second claim of Lemma 5.7, let T be the depth d parity decision tree obtained by starting at the root of the DAG Π, expanding the DAG into a tree, and removing any node at depth greater than d. We run Algorithm 3 on T and, by Lemma 5.10, the algorithm succeeds with probability q at least $\exp(-\Omega(d\beta + d^2(1-\alpha)) \geq S^{-O(1)}$, here we use the fact that $R \leq 1/100(1-\alpha)$. Since the DAG has size at most S, one of the nodes v must be reached, after a successful run, with probability at least $q/S \geq S^{-O(1)}$. By Lemma 5.9, this is also a lower bound on the probability that $\boldsymbol{x} \sim \prod_{i \in [k] \setminus B(M)} N^\cap(M, i)$ ends at this node. By Lemma 5.5, the rank of Ψ_v is bounded by $O(\log S)$.

Now fix some successful run of T that ends at v and apply Lemma 5.11 to find a corresponding F', M', ρ. Let $M^* = M \cup M'$ and let Π' be the affine-DAG consisting of the sub-DAG of Π of vertices reachable from v and restricted by ρ. Note that Lemma 5.11 ensures that ρ satisfies Ψ and thus the root of Π' is labeled by the empty system as desired. Furthermore, for any assignment to the blocks F', only selecting vertices within $N^\cap(M^*, [k] \setminus F')$, extending it to all blocks according to ρ can never introduce a non-edge outside of F'. Thus Π' solves $\text{NonEdge}_{\text{M}^*}$. Furthermore, note that v can not be a sink in Π since Π' solves a non-trivial problem. Thus v must have been reached after exactly d steps in T which means that Π' cannot have depth higher than $D - d$.

□

6 Lower Bound for Randomized Communication

In this section, we lower bound randomized communication cost required to solve $\mathsf{BClique}^{cc}_{\text{block}}(G, k)$.

Theorem 6.1. *For any integers n and k, and for any real $p \in [0, 1]$, if $\boldsymbol{G}$ is a graph sampled from $\mathcal{G}(n, p, k)$ then a.a.s. randomized communication protocol solving $\mathsf{BClique}^{cc}_{block}(\boldsymbol{G}, k)$ has cost*

$$\Omega\left(\min\left(\frac{1}{\sqrt{n^{0.05}(1-p)}}, \frac{n^{0.225}}{\log(kn)}\right)\right).$$

To see that Theorem 1.3 follows we set $p = 1 - n^{-1/3}$ and $k = n$. $\boldsymbol{G}$ then has $N = n^2$ vertices and we get the lower bound $\Omega(n^{0.22}) = N^{\Omega(1)}$ on the cost of the protocol as required.

For the rest of this section, recall that $S_G \coloneqq \mathsf{BClique}^{cc}_{\text{block}}(G,k) = \{(x,y,uv) \mid x,y \in \Sigma^k,\ uv \notin E(G),\ \{u,v\} \subseteq V(x,y)\}$, where $\Sigma = \{0,1\}^{\log n/2}$, $V(x,y) \in V^k$ is the list on nodes described by $x,y \in \Sigma^k$.

Theorem 6.2. *Let $s,k,n > 0$, and G be an s-almost-complete graph on k blocks of size n. Then the randomized communication complexity of S_G is $\Omega(\sqrt{n^{0.45}/s})$.*

Theorem 6.1 follows immediately from Theorem 6.2 and Lemma 3.2. To see this recall that Lemma 3.2 states that a.a.s. a graph $\boldsymbol{G}$ sampled as in Theorem 6.1 is s-almost-complete for $s = \max\{2\sqrt{n}(1-p), 9e^2\log(kn)\}$. From Theorem 6.2 it then follows that the randomized communication complexity of S_G is

$$\Omega\left(\sqrt{\frac{n^{0.45}}{s}}\right) = \Omega\left(\min\left(\frac{1}{\sqrt{n^{0.05}(1-p)}}, \frac{n^{0.225}}{\sqrt{\log(kn)}}\right)\right)$$

The rest of this section is dedicated to the proof of Theorem 6.2. By Yao's principle for randomized communication complexity [52], it suffices to show that every small cost deterministic protocol errs significantly on some input distribution. We will show this for the uniform distribution over inputs $(\boldsymbol{x},\boldsymbol{y}) \sim \Sigma^k \times \Sigma^k$.

6.1 Lifting Background

We start by introducing some concepts necessary to define subcube-like protocols, which we will use in our analysis. Every node v in a communication protocol Π is associated with a rectangle $R_v = X_v \times Y_v$. For a random variable $\boldsymbol{x}$ we write its min-entropy as $\mathrm{H}_\infty(\boldsymbol{x}) = \min_x \log(1/\Pr[\boldsymbol{x} = x])$. We say that a random variable $\boldsymbol{x} \in \{0,1\}^m$ is *γ-spread* if for every subset $I \subseteq [m]$, the marginal distribution $\boldsymbol{x}_I$ satisfies $\mathrm{H}_\infty(\boldsymbol{x}_I) \geq \gamma|I|$. We define *subcube-like communication protocols* [24, 46].

Definition 6.3. We say that a rectangle $R = X \times Y \subseteq \{0,1\}^m \times \{0,1\}^m$ is *γ-subcube-like* with respect to $I,J \subseteq [m]$ if X_I, Y_J are fixed to some values, and the random variables $\boldsymbol{x}_{[m]\setminus I}, \boldsymbol{y}_{[m]\setminus J}$ are γ-spread for $\boldsymbol{x} \sim X$ and $\boldsymbol{y} \sim Y$. We write $\mathsf{fix}(X) = I$ and $\mathsf{fix}(Y) = J$.

A communication protocol Π is called *γ-subcube-like* if all the rectangles associated to the nodes in Π are γ-subcube-like.

In our proof of Theorem 6.2, we use the following theorem from [24], which essentially says that, without loss of generality, we can assume the protocol to be subcube-like.

Theorem 6.4 ([24], Theorem 29)**.** *Let Π be a deterministic communication protocol with input from $X \times Y$. Then, for any constant $\gamma < 1$, there is a protocol Π' that γ-subcube-like, and has cost $|\Pi'| = O(|\Pi|/\varepsilon)$ such that $\Pr_{\boldsymbol{x},\boldsymbol{y}\sim X\times Y}[\Pi'(\boldsymbol{x},\boldsymbol{y}) \neq \Pi(\boldsymbol{x},\boldsymbol{y})] \leq \varepsilon$.*

6.2 Proof of Theorem 6.2

Assume that Π solves S_G with an error ε. By Theorem 6.4, there is a 0.9-subcube-like protocol Π' of cost $O(|\Pi|/\varepsilon) =: d$ solving S_G to an error 2ε. Sample uniform random inputs $\boldsymbol{x},\boldsymbol{y} \in \Sigma^k = \{0,1\}^{k\log|\Sigma|}$ for Alice and Bob. Let the output of $\Pi'(\boldsymbol{x},\boldsymbol{y})$ be an edge connecting nodes from blocks $\boldsymbol{a} \in [k]$ and $\boldsymbol{b} \in [k]$. We denote this edge with $(\boldsymbol{a}\boldsymbol{x}_{\boldsymbol{a}}\boldsymbol{y}_{\boldsymbol{a}}, \boldsymbol{b}\boldsymbol{x}_{\boldsymbol{b}}\boldsymbol{y}_{\boldsymbol{b}})$ meaning that both nodes are described by first specifying the block, and then specifying the number of the node inside the block

identified with $\Sigma \times \Sigma$. Let $\boldsymbol{\ell}$ be the leaf of Π' such that $(\boldsymbol{x}, \boldsymbol{y}) \in R_{\boldsymbol{\ell}}$. For a set $I \subseteq [k \log |\Sigma|]$ let the set $\mathsf{blocks}(I)$ be the set of all blocks $i \in [k]$ whose bits are mentioned in I.

The proof can be summarized as follows. If one of the blocks $\boldsymbol{a}$ and $\boldsymbol{b}$ is never mentioned in the fixed part of the leaf rectangle $R_{\boldsymbol{\ell}}$, then, by density of the graph, the answer is likely to be wrong. Otherwise, at least one bit describing the nodes in $\boldsymbol{a}$ and $\boldsymbol{b}$ is fixed at the leaf. In that case, we argue that since there are at most d blocks that can be mentioned, it is unlikely that a non-edge was discovered among those.

For an arbitrary node v of Π' let $D_v \coloneqq \mathsf{blocks}(\mathsf{fix}(X_v)) \cup \mathsf{blocks}(\mathsf{fix}(Y_v))$. Then the event where one of the block $\boldsymbol{a}, \boldsymbol{b}$ is never fixed in the run of the protocol is expressed simply as "$\{\boldsymbol{a}, \boldsymbol{b}\} \not\subseteq D_{\boldsymbol{\ell}}$". Then

$$\Pr[(\boldsymbol{a}\boldsymbol{x}_{\boldsymbol{a}}\boldsymbol{y}_{\boldsymbol{a}}, \boldsymbol{b}\boldsymbol{x}_{\boldsymbol{b}}\boldsymbol{y}_{\boldsymbol{b}}) \notin E(G)] \le \Pr[(\boldsymbol{a}\boldsymbol{x}_{\boldsymbol{a}}\boldsymbol{y}_{\boldsymbol{a}}, \boldsymbol{b}\boldsymbol{x}_{\boldsymbol{b}}\boldsymbol{y}_{\boldsymbol{b}}) \notin E(G) \mid \{\boldsymbol{a}, \boldsymbol{b}\} \not\subseteq D_{\boldsymbol{\ell}}] \tag{5}$$
$$+ \Pr[(\boldsymbol{a}\boldsymbol{x}_{\boldsymbol{a}}\boldsymbol{y}_{\boldsymbol{a}}, \boldsymbol{b}\boldsymbol{x}_{\boldsymbol{b}}\boldsymbol{y}_{\boldsymbol{b}}) \notin E(G) \wedge \{\boldsymbol{a}, \boldsymbol{b}\} \subseteq D_{\boldsymbol{\ell}}] \tag{6}$$

Bounding the summand (5). Observe that for each leaf ℓ of Π' the probability $\Pr[\{\boldsymbol{a}, \boldsymbol{b}\} \subseteq D_{\boldsymbol{\ell}} \mid \boldsymbol{\ell} = \ell] \in \{0, 1\}$, i.e., leaves can be categorized into safe and dangerous: ℓ is *safe* if $\Pr[\{\boldsymbol{a}, \boldsymbol{b}\} \subseteq D_{\boldsymbol{\ell}} \mid \boldsymbol{\ell} = \ell] = 0$, otherwise it is *dangerous*. We also note the following useful claim.

Claim 6.5. *Suppose ℓ is a safe leaf, then*

$$\Pr[(\boldsymbol{a}\boldsymbol{x}_{\boldsymbol{a}}\boldsymbol{y}_{\boldsymbol{a}}, \boldsymbol{b}\boldsymbol{x}_{\boldsymbol{b}}\boldsymbol{y}_{\boldsymbol{b}}) \notin E(G) \mid (\boldsymbol{x}, \boldsymbol{y}) \in R_\ell] \le s/n^{0.45}. \tag{7}$$

Proof of Claim 6.5. Wlog assume that $\boldsymbol{a} \notin D_{\boldsymbol{\ell}}$. Fix $\boldsymbol{x}_{\boldsymbol{a}}\boldsymbol{x}_{\boldsymbol{b}}\boldsymbol{y}_{\boldsymbol{b}} = \alpha \in \Sigma^3$ arbitrarily. By the total probability law it suffices to bound the target probability conditioned on each value α. By definition of $\mathsf{blocks}(\mathsf{fix}(Y_{\boldsymbol{\ell}}))$ we have $\mathrm{H}_\infty(\boldsymbol{y}_{\boldsymbol{a}} \mid (\boldsymbol{x}, \boldsymbol{y}) \in R_\ell) \ge 0.9 \log |\Sigma|$. By s-neighbor density we have that $\Pr[(\boldsymbol{a}\boldsymbol{x}_{\boldsymbol{a}}\boldsymbol{y}_{\boldsymbol{a}}, \boldsymbol{b}\boldsymbol{x}_{\boldsymbol{b}}\boldsymbol{y}_{\boldsymbol{b}}) \notin E(G) \mid (\boldsymbol{x}, \boldsymbol{y}) \in R_\ell, \boldsymbol{x}_{\boldsymbol{a}}\boldsymbol{x}_{\boldsymbol{b}}\boldsymbol{y}_{\boldsymbol{b}} = \alpha] \le |\Sigma|^{-0.9} s \le n^{-0.45} s$. □

Since (5) is a convex combination of left-hand sides of (7), we conclude by Claim 6.5 that the summand (5) is bounded with $s/n^{0.45}$ as well.

Bounding the summand (6). We would like to apply a union bound over all values of $\{\boldsymbol{a}, \boldsymbol{b}\}$. We first show this for a *fixed* pair of blocks a and b:

Claim 6.6. *For every $a \neq b \in [k]$ we have*

$$\Pr[(a\boldsymbol{x}_a\boldsymbol{y}_a, b\boldsymbol{x}_b\boldsymbol{y}_b) \notin E(G) \mid \{a, b\} \subseteq D_{\boldsymbol{\ell}}] \le s/n^{0.45}.$$

Proof. Let the run of the protocol $\Pi'(\boldsymbol{x}, \boldsymbol{y})$, be the sequence $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d = \boldsymbol{\ell}$ of nodes in the protocol tree.

Consider a node v in Π' corresponding to a rectangle $R_v = X_v \times Y_v \ni (\boldsymbol{x}, \boldsymbol{y})$ such that $\{a, b\} \subseteq D_v$, but the inclusion is false for the parent of v in Π'. Let T_i be the set of these nodes in Π' at the depth i, and $T \coloneqq \bigcup_{i \le d} T_i$. We then have

$$\Pr[(a\boldsymbol{x}_a\boldsymbol{y}_a, b\boldsymbol{x}_b\boldsymbol{y}_b) \notin E(G) \mid \{a, b\} \subseteq D_{\boldsymbol{\ell}}] = \sum_{i \in [k], v \in T_i} \Pr[\boldsymbol{v}_i = v] \Pr[(a\boldsymbol{x}_a\boldsymbol{y}_a, b\boldsymbol{x}_b\boldsymbol{y}_b) \notin E(G) \mid (\boldsymbol{x}, \boldsymbol{y}) \in R_v].$$

Then it suffices to bound the probability $\Pr[(a\boldsymbol{x}_a\boldsymbol{y}_a, b\boldsymbol{x}_b\boldsymbol{y}_b) \notin E(G) \mid (\boldsymbol{x},\boldsymbol{y}) \in R_v]$ for an arbitrary $v \in T$, since the target probability is a convex combination of these probabilities. Now suppose wlog that the node v was a node where Alice spoke, i.e., for the parent u of v we have $Y_u = Y_v$. Then $\{a,b\} \not\subseteq \mathsf{blocks}(\mathsf{fix}(Y_v))$. Indeed, since $v \in T$ we have $\{a,b\} \not\subseteq D_u$, so $\{a,b\} \not\subseteq \mathsf{blocks}(\mathsf{fix}(Y_u)) = \mathsf{blocks}(\mathsf{fix}(Y_v))$. Assume again wlog that $b \notin \mathsf{blocks}(\mathsf{fix}(Y_v))$.

Finally, we have by the subcube likeness of R_v and the definition of $\mathsf{blocks}(\mathsf{fix}(Y_v))$ that $\mathrm{H}_\infty(\boldsymbol{y}_b \mid \boldsymbol{y} \in Y_v) \geq 0.9 \log|\Sigma|$. Then by the graph density $\Pr[(a\boldsymbol{x}_a\boldsymbol{y}_a, b\boldsymbol{x}_b\boldsymbol{y}_b) \notin E(G) \mid (\boldsymbol{x},\boldsymbol{y}) \in R_v] \leq s/n^{0.45}$ as required. □

We need some additional notation to implement our union bound. For a protocol leaf ℓ define

$$p_{a,b}(\ell) \coloneqq \Pr_{\boldsymbol{z},\boldsymbol{w}\sim R_\ell}[(a\boldsymbol{z}_a\boldsymbol{w}_a, b\boldsymbol{z}_b\boldsymbol{w}_b) \notin E(G)].$$

Since the pair $(\boldsymbol{a},\boldsymbol{b})$ is uniquely determined by $\boldsymbol{\ell}$, so denote $a(\ell)$ and $b(\ell)$ the blocks returned in the leaf ℓ. Having that, we rewrite

$$\begin{aligned}
&\Pr[(\boldsymbol{a}\boldsymbol{x}_{\boldsymbol{a}}\boldsymbol{y}_{\boldsymbol{a}}, \boldsymbol{b}\boldsymbol{x}_{\boldsymbol{b}}\boldsymbol{y}_{\boldsymbol{b}}) \notin E(G) \wedge \{\boldsymbol{a},\boldsymbol{b}\} \subseteq D_{\boldsymbol{\ell}}] \\
\text{(def. of dangerous leaves)} \quad &= \sum_{\ell\colon \text{ dangerous leaf}} p_{a(\ell),b(\ell)}(\ell)\Pr[(\boldsymbol{x},\boldsymbol{y}) \in R_\ell] \\
&= \mathbb{E}[p_{\boldsymbol{a},\boldsymbol{b}}(\boldsymbol{\ell}) \cdot \mathbb{1}_{\{\boldsymbol{a},\boldsymbol{b}\}\subseteq D_{\boldsymbol{\ell}}}] \\
\text{(sum includes the term } \boldsymbol{a},\boldsymbol{b}) \quad &\leq \sum_{a\neq b\in[k]} \mathbb{E}[\mathbb{1}_{\{a,b\}\subseteq D_{\boldsymbol{\ell}}} \cdot p_{a,b}(\boldsymbol{\ell})] \\
&= \sum_{a\neq b\in[k]} \Pr[\{a,b\} \subseteq D_{\boldsymbol{\ell}} \wedge (a\boldsymbol{x}_a\boldsymbol{y}_a, b\boldsymbol{x}_b\boldsymbol{y}_b) \notin E(G)] \\
&= \sum_{a\neq b\in[k]} \Pr[(a\boldsymbol{x}_a\boldsymbol{y}_a, b\boldsymbol{x}_b\boldsymbol{y}_b) \notin E(G) \mid \{a,b\} \subseteq D_{\boldsymbol{\ell}}] \cdot \Pr[\{a,b\} \subseteq D_{\boldsymbol{\ell}}] \\
\text{(number of pairs is } \leq |D_{\boldsymbol{\ell}}|^2) \quad &\leq \mathbb{E}[|D_{\boldsymbol{\ell}}|^2] \cdot \max_{a\neq b\in[k]} \Pr[(a\boldsymbol{x}_a\boldsymbol{y}_a, b\boldsymbol{x}_b\boldsymbol{y}_b) \notin E(G) \mid \{a,b\} \subseteq D_{\boldsymbol{\ell}}] \\
\text{(Claim 6.6)} \quad &\leq d^2 s/n^{0.45}.
\end{aligned}$$

Finally, we have established that the success probability of Π' is at most $2d^2s/n^{0.45} \geq 1 - 2\varepsilon = \Omega(1)$, so $d = \Omega(\sqrt{n^{0.45}/s})$ as required.

Acknowledgements

We would like to thank Dmitry Itsykson and Kilian Risse for insightful discussions.

David Engström and Susanna F. de Rezende received funding from the Knut and Alice Wallenberg grant KAW 2023.0116, ELLIIT, and the Swedish Research Council grant 2021-05104. Yassine Ghannane and Duri Andrea Janett were supported by the Independent Research Fund Denmark grant 9040-00389B. Artur Riazanov is supported by Swiss State Secretariat for Education, Research, and Innovation (SERI) under contract number MB22.00026. We also gratefully acknowledge that we have benefited greatly from being part of Basic Algorithms Research Copenhagen (BARC) environment financed by the Villum Investigator grant 54451.

References

[1] Yaroslav Alekseev and Dmitry Itsykson. Lifting to bounded-depth and regular resolutions over parities via games. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC '25)*, pages 584–595, 2025. doi:10.1145/3717823.3718150.

[2] Albert Atserias, Ilario Bonacina, Susanna F. de Rezende, Massimo Lauria, Jakob Nordström, and Alexander Razborov. Clique is hard on average for regular resolution. *Journal of the ACM*, 68(4):23:1–23:26, 2021. Preliminary version in *STOC '18*. doi:10.1145/3449352.

[3] Paul Beame, Russell Impagliazzo, and Ashish Sabharwal. The resolution complexity of independent sets and vertex covers in random graphs. *Computational Complexity*, 16(3):245–297, 2007. Preliminary version in *CCC '01*. doi:10.1007/S00037-007-0230-0.

[4] Paul Beame, Henry Kautz, and Ashish Sabharwal. Towards understanding and harnessing the potential of clause learning. *Journal of Artificial Intelligence Research*, 22(1):319–351, 2004. doi:10.1613/jair.1410.

[5] Paul Beame, Toniann Pitassi, and Nathan Segerlind. Lower bounds for Lovász–Schrijver systems and beyond follow from multiparty communication complexity. *SIAM Journal on Computing*, 37(3):845–869, 2007. Preliminary version in *ICALP '05*. doi:10.1137/060654645.

[6] Paul Beame and Michael Whitmeyer. Multiparty communication complexity of collision-finding and cutting planes proofs of concise pigeonhole principles. In *Procceedings of the 52nd International Colloquium on Automata, Languages, and Programming (ICALP '25)*, volume 334 of *LIPIcs*, pages 21:1–21:20, 2025. doi:10.4230/LIPIcs.ICALP.2025.21.

[7] Olaf Beyersdorff, Nicola Galesi, and Massimo Lauria. Parameterized complexity of DPLL search procedures. *ACM Transactions on Computational Logic*, 14(3):20:1–20:21, 2013. Preliminary version in *SAT '11*. doi:10.1145/2499937.2499941.

[8] Olaf Beyersdorff, Nicola Galesi, Massimo Lauria, and Alexander A Razborov. Parameterized bounded-depth Frege is not optimal. *ACM Transactions on Computation Theory (TOCT)*, 4(3):1–16, 2012. Preliminary version in *ICALP '11*. doi:10.1145/2355580.2355582.

[9] Sreejata Kishor Bhattacharya and Arkadev Chattopadhyay. Exponential lower bounds on the size of reslin proofs of nearly quadratic depth. *arXiv preprint arXiv:2507.23008*, 2025. doi:10.48550/ARXIV.2507.23008.

[10] Sreejata Kishor Bhattacharya, Arkadev Chattopadhyay, and Pavel Dvorák. Exponential separation between powers of regular and general resolution over parities. In *Proceedings of the 39th Computational Complexity Conference (CCC '24)*, volume 300 of *LIPIcs*, pages 23:1–23:32, 2024. doi:10.1145/3618260.3649652.

[11] Sam 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, 2021. doi:10.3233/faia200990.

[12] Farzan Byramji and Russell Impagliazzo. Lower bounds for bit pigeonhole principles in bounded-depth resolution over parities. *arXiv preprint arXiv:2511.20023*, 2025. doi:10.48550/ARXIV.2511.20023.

[13] Marco L. Carmosino, Jiawei Gao, Russell Impagliazzo, Ivan Mihajlin, Ramamohan Paturi, and Stefan Schneider. Nondeterministic extensions of the strong exponential time hypothesis and consequences for non-reducibility. In *Proceedings of the 7th ACM Conference on Innovations in Theoretical Computer Science (ITCS '16)*, page 261–270, 2016. doi:10.1145/2840728.2840746.

[14] Arkadev Chattopadhyay, Shachar Lovett, and Marc Vinyals. Equality alone does not simulate randomness. In *Proceedings of the 34th Computational Complexity Conference (CCC '19)*, pages 14:1–14:11, 2019. doi:10.4230/LIPIcs.CCC.2019.14.

[15] Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Linear FPT reductions and computational lower bounds. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC '04)*, pages 212–221, 2004. doi:10.1145/1007352.1007391.

[16] Tsun-Ming Cheung, Hamed Hatami, Kaave Hosseini, Aleksandar Nikolov, Toniann Pitassi, and Morgan Shirley. A lower bound on the trace norm of boolean matrices and its applications. In *Proceedings of the 16th Innovations in Theoretical Computer Science Conference (ITCS '25)*, pages 37:1–37:15, 2025. doi:10.4230/LIPIcs.ITCS.2025.37.

[17] Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC '96)*, page 174–183, 1996. doi:10.1145/237814.237860.

[18] William Cook, Collette Rene Coullard, and György Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, 1987. doi:10.1016/0166-218x(87)90039-4.

[19] Stefan Dantchev, Nicola Galesi, Abdul Ghani, and Barnaby Martin. Proof complexity and the binary encoding of combinatorial principles. *SIAM Journal on Computing*, 53(3):764–802, 2024. doi:10.1137/20M134784X.

[20] Susanna de Rezende, Aaron Potechin, and Kilian Risse. Clique is hard on average for unary Sherali-Adams. In *Proceedings of the 64th IEEE Annual Symposium on Foundations of Computer Science (FOCS 23')*, pages 12–25, 2023. doi:10.1109/focs57990.2023.00008.

[21] Klim Efremenko, Michal Garlík, and Dmitry Itsykson. Lower bounds for regular resolution over parities. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24)*, pages 640–651, 2024. doi:10.1145/3618260.3649652.

[22] Klim Efremenko and Dmitry Itsykson. Amortized closure and its applications in lifting for resolution over parities. In *Proceedings of the 40th Computational Complexity Conference (CCC '25)*, pages 8:1–8:24, 2025. doi:10.4230/LIPICS.CCC.2025.8.

[23] 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*. doi:10.4086/TOC.2020.V016A013.

[24] Mika Göös, Tom Gur, Siddhartha Jain, and Jiawei Li. Quantum communication advantage in TFNP. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, (STOC '25)*, pages 1465–1475, 2025. doi:10.1145/3717823.3718155.

[25] Mika Göös, Nathaniel Harms, and Artur Riazanov. Equality is far weaker than constant-cost communication. In *Proceedings of the 29th International Conference on Randomization and Computation (RANDOM '25)*, volume 353, pages 58:1–58:14, 2025. doi:10.4230/LIPIcs.APPROX/RANDOM.2025.58.

[26] Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. *SIAM Journal on Computing*, 49(4):FOCS17–441–FOCS17–461, 2020. doi:10.1137/17M115339X.

[27] Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. *SIAM Journal on Computing*, 47(5):1778–1806, 2018. Preliminary version in *STOC '14*. doi:10.1137/16M1082007.

[28] Armin Haken and Stephen A. Cook. An exponential lower bound for the size of monotone real circuits. *Journal of Computer and System Sciences*, 58(2):326–335, 1999. doi:10.1006/jcss.1998.1617.

[29] Johan Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182(1), 1999. doi:DOI:10.1007/BF02392825.

[30] Pavel Hrubeš and Pavel Pudlák. A note on monotone real circuits. *Information Processing Letters*, 131:15–19, 2018. doi:10.1016/j.ipl.2017.11.002.

[31] Trinh Huynh and Jakob Nordström. On the virtue of succinct proofs: amplifying communication complexity hardness to time-space trade-offs in proof complexity. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing Conference, (STOC '12)*, pages 233–248, 2012. doi:10.1145/2213977.2214000.

[32] Russell Impagliazzo, Toniann Pitassi, and Alasdair Urquhart. Upper and lower bounds for tree-like cutting planes proofs. In *Proceedings 9th Annual IEEE Symposium on Logic in Computer Science (LICS '94)*, pages 220–228, 1994. doi:10.1109/LICS.1994.316069.

[33] Dmitry Itsykson and Artur Riazanov. Proof complexity of natural formulas via communication arguments. In *Proceedings of the 36th Computational Complexity Conference (CCC '21)*, pages 3:1–3:34, 2021. doi:10.4230/LIPIcs.CCC.2021.3.

[34] Dmitry Itsykson and Dmitry Sokolov. Resolution over linear equations modulo two. *Annals of Pure and Applied Logic*, 171(1):102722, 2020. Preliminary version in *MFCS '14*. doi:10.1016/J.APAL.2019.102722.

[35] Stasys Jukna. Clique problem, cutting plane proofs and communication complexity. *Information Processing Letters*, 112(20):772–777, 2012. doi:10.1016/j.ipl.2012.07.003.

[36] Richard M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, 1972. doi:10.1007/978-1-4684-2001-2_9.

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

[38] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997. doi:10.1016/S0065-2458(08)60342-3.

[39] Jean B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001. doi:10.1137/s1052623400366802.

[40] Massimo Lauria. Cliques enumeration and tree-like resolution proofs. *Information Processing Letters*, 135:62–67, 2018. doi:10.1016/J.IPL.2018.03.001.

[41] Massimo Lauria, Pavel Pudlák, Vojtěch Rödl, and Neil Thapen. The complexity of proving that a graph is Ramsey. *Combinatorica*, 37(2):253–268, 2017. Preliminary version in *ICALP '13*. doi:10.1007/S00493-015-3193-9.

[42] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge University Press, 2017. doi:10.1017/CBO9780511813603.

[43] Shuo Pang. Large clique is hard on average for resolution. In *International Computer Science Symposium in Russia (CSR '21)*, pages 361–380, 2021. doi:10.1007/978-3-030-79416-3_22.

[44] Anup Rao and Amir Yehudayoff. *Communication Complexity and Applications*. Cambridge University Press, 2020. doi:10.1145/3357713.3384286.

[45] Alexander A. Razborov. Lower bounds on the monotone complexity of some boolean functions. *Doklady Akademii Nauk SSSR*, 1985. doi:10.1007/bf01157687.

[46] Artur Riazanov, Anastasia Sofronova, Dmitry Sokolov, and Weiqiang Yuan. Searching for falsified clause in random $(\log n)$-CNFs is hard for randomized communication. In *Proceedings of the 29th International Conference on Randomization and Computation (RANDOM '25)*, pages 64:1–64:17, 2025. doi:10.4230/LIPIcs.APPROX/RANDOM.2025.64.

[47] Benjamin Rossman. On the constant-depth complexity of k-clique. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC '08)*, pages 721–730, 2008. doi:10.1145/1374376.1374480.

[48] Benjamin Rossman. The monotone complexity of k-clique on random graphs. *SIAM Journal on Computing*, 43(1):256–279, 2014. Preliminary version in *FOCS '10*. doi:10.1137/110839059.

[49] 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, 2017. doi:10.1007/978-3-319-58747-9_26.

[50] 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, 2024. doi:10.1145/3618260.3649636.

[51] Guangxu Yang and Jiapeng Zhang. Communication lower bounds for collision problems via density increment arguments. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24)*, pages 630–639, 2024. doi:10.1145/3618260.3649607.

[52] Andrew C. Yao. Lower bounds by probabilistic arguments. In *Proceedings of the 24th Annual Symposium on Foundations of Computer Science (FOCS '83)*, pages 420–428, 1983. doi:10.1109/SFCS.1983.30.

[53] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(6):103–128, 2007. doi:10.4086/TOC.2007.V003A006.

Paper C

Communication Lower Bounds Beyond Jukna's Criterion

Communication Lower Bounds Beyond Jukna's Criterion

Duri Andrea Janett
University of Copenhagen

Robert Robere
McGill

Anastasia Sofronova
EPFL

Dmitry Sokolov
Université de Montréal

Abstract

The Cutting Planes proof system is among the weakest proof systems for which no lower bounds are known for pseudorandom generator tautologies. We consider a problem intermediate between proving lower bounds for random $\Theta(\log n)$-CNF formulas, and proving lower bounds on inner-product Nisan–Wigderson generators for cutting planes. Previous techniques based on Jukna's criterion do not apply to this problem. We prove a rectangle-DAG lower bound for the intermediate problem, using a novel technique based on Razborov's approximation method.

1 Introduction

The *pseudorandom generator tautologies*, introduced independently by Krajíček [Kra01] and Alekhnovich, Ben-Sasson, Razborov, and Wigderson [ABRW04], represent some of the most important families of tautologies studied in proof complexity. Informally, given a candidate pseudorandom generator $G\colon \{0,1\}^n \to \{0,1\}^m$, the pseudorandom generator tautologies for G assert the statement that "$b \notin \mathrm{Im}(G)$" for all strings b. Considering G as a candidate pseudorandom generator, a short proof that a fixed string b is not in the range of G would imply that the output of G cannot be a random string, and hence represents an "**NP**-certificate" that G is *not* pseudorandom. However, the study of pseudorandom generator tautologies turns out to be motivated for a variety of reasons in complexity theory and beyond (we refer to [ABRW04; Kra01; Raz15] for a more detailed discussion), such as:

- **Truth-Table Generators.** Consider when $G := \mathrm{Eval}_t\colon \{0,1\}^t \to \{0,1\}^{2^n}$ takes as input a boolean circuit described by a string of length t, and outputs the truth-table of the function encoded by the circuit. Proving that any particular string $f_n \notin \mathrm{Im}(G)$ is tantamount to proving a circuit lower bound for f_n, and hence one can use these tautologies to encode problems like the **NP** vs. **P**/poly problem in propositional proof complexity.
- **Candidate Hard Tautologies.** While researchers have been successful in proving strong lower bounds for weak proof systems, proving lower bounds for *strong* proof systems like Frege or Extended Frege (even conditionally) is viewed as completely out of reach. A major issue with studying stronger proof systems is that we lack good candidate hard tautologies for these problems, with many prior candidates having been shown to be easy for these proof systems. Appropriately chosen pseudorandom generator tautologies have been put forward as candidate hard examples for *any* proof system, see [ABRW04; Raz15] for more details.

An important instantiation of the pseudorandom generator methodology are the *Nisan–Wigderson*-style pseudorandom generator tautologies [ABRW04], which are defined as follows. Fix a bipartite *dependency graph* $G = (L, R, E)$, with $L = [m]$, $R = [n]$, and in which each vertex in L has degree Δ, and let $\vec{f} = (f_1, f_2, \dots, f_m)$ be a list of boolean functions $f_i\colon \{0,1\}^\Delta \to \{0,1\}$. For any $i \in L$, let $N_L(i) \subseteq R$ denote the neighbours of i. We associate each vertex $i \in L$ with the function f_i, and each vertex $j \in R$ with an input variable $x_j \in \{0,1\}$, and define the function

$$\mathcal{F}_{\vec{f},G}\colon x \mapsto (f_1(x \restriction_{N_L(1)}), f_2(x \restriction_{N_L(2)}), \dots, f_m(x \restriction_{N_L(m)})).$$

The tautologies then assert, as before, that an arbitrary string $b \notin \mathrm{Im}(\mathcal{F}_{\vec{f},G})$. These generators have been particularly well-studied in the literature as a concrete instantiations of PRG tautologies, for particular choices of hard functions $\vec{f}$ and where the underlying graph G is an expander graph [ABRW04; Raz15; Sok22]. Notably, Razborov [Raz15] outlined some explicit conjectures regarding the hardness of (certain versions) of these formulas for Frege and Extended Frege. Broadly speaking, he conjectured that if the functions $\vec{f}$ are *hard to compute* by some class of functions $\mathcal{C}$ on average (e.g. boolean formulas or boolean circuits, in the case of Frege or Extended Frege), then the tautologies associated with $\mathcal{F}_{\vec{f},G}$ should be hard to refute by proofs whose lines are drawn from $\mathcal{C}$. In certain parameter regimes and encodings, lower bounds for the Nisan–Wigderson generators are known for Resolution and Polynomial Calculus [ABRW04], as well as the Res(k) proof systems [Raz15; Sok22]. Recent work has also shown new lower bounds for the PCR and Sherali-Adams proof systems, via a variant principle [GGRT26].

We further note that the Nisan–Wigderson generator tautologies are spiritually similar to *random unsatisfiable k-CNFs*, which are another often-conjectured candidate family of hard formulas for all propositional proof systems [CS88; BP96; BI99; ABE02; SBI04; HP17; FPPR22]. For integer parameters m, n, k, let $F(m, n, k)$ denote the distribution of k-CNF formulas obtained by randomly sampling m uniformly random clauses of width k over n variables with replacement. One can view a sample from this distribution as an instance of the Nisan–Wigderson generator tautologies $\mathcal{F}_{\vec{f},G}$, where $G = ([m] \cup [n], E)$ is a random bipartite graph in which every left vertex has k randomly chosen neighbours (with replacement), and each function f_i is a uniformly random clause of width k. When m is sufficiently large (say, $m \geq \Omega(n2^k)$), it is known that a sample from this distribution is unsatisfiable with high probability. Since the functions $\vec{f}$ are very simple — being random clauses — for these formulas, the hardness must come from the structure of the clauses (typically, boundary expansion being the most significant property exploited). However, when m is sufficiently large in n and k, the formulas become easy to refute (even for tree-like Resolution [BP96]), and so these examples cannot be generator tautologies for any proof system in the large stretch regime $m \gg 2^{n^\epsilon}$.

Nisan–Wigderson Generators and Cutting Planes Proofs. The *Cutting Planes* proof system is a notable proof system for which we are currently *unable* to prove strong lower bounds for any PRG tautologies, although there is a rich and well-developed literature of proving lower bounds for other formulas [BPR97; Kra97; Pud97; HP17; FPPR22; Sok24]. Of course, in order to apply the paradigm to Cutting Planes, we first need to choose a candidate hard sequence of functions $\vec{f}$. Recent work has proved strong lower bounds on Cutting Planes refutations for random Δ-width CNFs when $\Delta = \Theta(\log n)$ [HP17; FPPR22; Sok24], although these lower bounds only hold when $m = O(n2^\Delta)$. This is a starting point, but to prove lower bounds for larger stretch, we must consider a different family of hard functions.

Since the lines of Cutting Planes proofs are integer-linear inequalities, we could follow Razborov's suggestion above for Frege and Extended Frege, and select a family of functions which are hard to represent by such lines. A natural choice would be parity functions, and indeed, in the original paper introducing the PRG tautologies, Alekhnovich et al introduced the *Nisan generator tautologies*, which are the restriction of the Nisan–Wigderson generators where the hard functions $\vec{f}$ are each chosen to be *parities* of their input bits [ABRW04]. The tautologies are essentially an overdetermined unsatisfiable system of $\mathbb{F}_2$-linear equations, and these were shown to be hard for Resolution in the right parameter regimes [ABRW04]. However, despite the fact that integer-linear inequalities cannot represent $\mathbb{F}_2$-linear equations, it is known that Cutting Planes has *quasipolynomial*-size refutations of any unsatisfiable system of $\mathbb{F}_2$-linear equations [DT20], and hence the Nisan generator tautologies cannot be exponentially hard for Cutting Planes.

The next obvious candidate would be to build Nisan–Wigderson generators using the *inner product* function $\mathsf{IP}_\Delta(x, y)$: $\{0,1\}^\Delta \times \{0,1\}^\Delta \to \{0,1\}$ defined by

$$\mathsf{IP}(x, y) = \sum_{i=1}^{\Delta} x_i y_i \pmod 2,$$

defined on a suitable expander graph G. However, the existing lower bound techniques for Cutting Planes run immediately into technical issues when adapting the techniques to these formulas. All existing lower bound techniques for Cutting Planes proofs prove lower bounds for the stronger models of *triangle dags*

and *rectangle dags* in communication complexity[1] (see Section 2 for formal definitions), via either *query-to-communication lifting theorems* [GGKS20], or directly via feasible-interpolation [BPR97; Kra97; Pud97] and related techniques [HP17; FPPR22; Sok17; Sok24]. It is natural to evaluate each of these techniques, and see their limitations when applied to Nisan–Wigderson generators.

First, the classical feasible interpolation techniques, as used by Pudlák in proving the first exponential-size Cutting Planes lower bounds [Pud97], only apply to *split* formulas of the form $F(x, y, z) = G(x, z) \wedge H(y, z)$, where G and H are CNF formulas that only share a set of common z variables. In the Nisan–Wigderson generators, there are no such natural partitions to choose for these formulas, and hence it is difficult to even begin applying the technique.

Second, we could resolve this problem if we could prove a query-to-communication lifting theorem for dag-like communication using the Inner Product as the inner gadget. Such lifting theorems exist for *tree-like* communication complexity [CFKMP19], but proving such a lifting theorem for dag-like communication remains an open problem. Indeed, currently all known proofs of the lifting theorem for dag-like communication protocols use the Index gadget [GGKS20; LMMPZ22].

Third, and perhaps most promising, one could hope to follow the recent work which has proved lower bounds for random $\Theta(\log n)$-CNFs [HP17; FPPR22; Sok24]. These proceed by generalizing the classical feasible interpolation techniques to apply to arbitrary formulas, and then applying techniques from monotone circuit complexity. Hrubés and Pudlák [HP17] and independently Fleming, Pankratov, Pitassi, and Robere [FPPR22], adapt the *monotone lower bounds criterion* of Jukna [Juk97], while Sokolov [Sok24] gave a simpler proof by adapting the *bottleneck counting method* directly to dag-like communication [Hak95].

However, if we wished to adapt this proof technique to Nisan–Wigderson generators using constraints other than width-$\Theta(\log n)$ clauses, we run into immediate issues. Most importantly, examining the proofs of [HP17; FPPR22; Sok24], the following "density property" of clauses is crucial. If $C(x, y)$ is a clause with variables partitioned arbitrarily into two sets X and Y, and x^* is any assignment to the X variables, then $|\{y : C(x^*, y) = 0\}| \leq 1/2^{|Y|}$. Of particular importance here is the exponential dependence on $|Y|$, and it is easy to see that the analogous claim fails for $\mathrm{IP}(x, y)$: given any assignment x^*, we have $|\{y : \mathrm{IP}(x^*, y) = 0\}| = 1/2$.

Our Contribution. In this paper, we study an intermediate problem which we regard as "between" that of proving lower bounds for random $\Theta(\log n)$-CNFs and the inner-product Nisan–Wigderson generator. On the one hand, we draw our constraints randomly with replacement, just as in the distribution $\mathcal{F}(m, n, \Delta)$ of random Δ-CNF formulas. On the other, our intermediate problem shares the same property as the inner-product Nisan–Wigderson generator, in that neither the bottleneck counting arguments nor Jukna's criterion imply lower bounds for our generator as the density property discussed just above fails.

Before stating our main theorem, we must introduce the distribution on formulas we study (cf. Section 4). Recall the distribution $\mathcal{F}(m, n, \Delta)$ of random Δ-CNF formulas with m clauses over n variables. Analogously, define the distribution $\mathcal{P}(m, n, \Delta)$ which selects m random parity constraints of width Δ (with replacement) from a pool of n variables. Define the distribution $(\mathcal{F} \otimes \mathcal{P})(m, n, \Delta)$ as follows: sample $F(x) = \bigwedge_i C_i(x) \sim \mathcal{F}(m, n, \Delta)$ and $G(y) = \bigwedge_i P_i(y) \sim \mathcal{P}(m, n, \Delta)$ on disjoint sets of variables x and y. Then output the formula $\bigwedge_i (C_i(x) \vee P_i(y))$.

One can easily see that the same density property fails for formulas drawn from $\mathcal{F} \otimes \mathcal{P}$: if x^* is any assignment to the x variables and $C_i(x) \vee P_i(y)$ is a constraint, then $|\{y : C_i(x^*) \vee P_i(y) = 0\}| = 1/2$, just as with $\mathrm{IP}(x, y)$. Hence, both the bottleneck counting techniques of Sokolov [Sok24] and the monotone lower-bounds criterion of Jukna [Juk97] fails. However, we are still able to prove strong lower bounds on this distribution:

Theorem 1.1. Let $\alpha, d > 0$, be constants and $\Delta = d \log n$, $m = \alpha n 2^{\Delta}$. For $\varphi \sim (\mathcal{F} \otimes \mathcal{P})(m, n, \Delta)$, a rectangle-DAG Π solving $\mathrm{Search}(\varphi)$ has length $|\Pi|$ at least $2^{\Omega(n)}$ with high probability.

[1] In the classical literature, these techniques were formalized using feasible interpolation, as a reduction to monotone circuit lower bounds.

As an immediate corollary from known results [BPR97; Kra97], the above theorem implies the same lower bounds for "unary" Cutting Planes proofs, where all intermediate coefficients in the proofs are bounded.

We prove our lower bounds by adapting a different monotone circuit lower bound technique directly to rectangle dags: the classical *method of approximations*, due to Razborov [Raz85], particularly utilizing the *sunflower lemma*. We believe this new proof technique to be of independent interest. Technically speaking, we must overcome several hurdles in order to translate the method of approximations directly to these formulas: for instance, we must introduce notions of approximators for rectangles that mirror the classical definitions, and show how to apply a version of Razborov's sunflower plucking procedure to them. We find it more convenient to work with the notion of *robust sunflowers* introduced by Rossman [Ros10], and particularly employing the parameters in the improved sunflower lemma proved by [ALWZ21].

2 Preliminaries

We use the symbol $\sqcup$ to denote a disjoint union of sets. A probability distribution $\mathcal{D}$ over $\{0,1\}^n$ is said to be *r-wise independent* if every restriction of $\mathcal{D}$ to r coordinates is uniform on $\{0,1\}^r$.

For a graph $G = (V, E)$ and a subset $S \subseteq V$, we write $N(S)$ for the neighbors of S, and $\partial(S)$ for the unique neighbors of S, i.e., $\partial(S) = \{v \in V \mid \exists! u \in S \colon \{v, u\} \in E\}$. For a *directed acyclic graph (DAG)* $G = (V, A)$, we call $u \in V$ an *out-neighbor of* v if $(v, u) \in A$, and we call $u \in V$ an *in-neighbor of* v if $(u, v) \in A$. For a node $v \in V$, we let $\mathrm{desc}(v)$ be the set of all vertices $u \in V$ such that there is a directed path from v to u.

For a subset $S \subseteq X \times Y$, we write $S\restriction_X = \{x \in X \mid \exists y \in Y \text{ s.t. } (x, y) \in S\}$ and similarly $S\restriction_Y = \{y \in Y \mid \exists x \in X \text{ s.t. } (x, y) \in S\}$. A subset $R \subseteq X \times Y$ with $R = R\restriction_X \times R\restriction_Y$ is a *rectangle*. A rectangle is said to be *monochromatic* for a clause C, or *C-monochromatic*, if for all $(x, y) \in R$, $C(x, y) = 0$.

Rectangle-DAGs. For a given unsatisfiable CNF formula F and a partition of its variables $\mathrm{Vars}(F) = V_X \sqcup V_Y$, we consider the *falsified clause search problem* $\mathsf{Search}(F) \subseteq (X \times Y) \times F$, where X and Y are the sets of assignments to V_X and V_Y, respectively, and $(x, y, C) \in \mathsf{Search}(F)$ if and only if the assignment $x \cup y$ falsifies the clause C.

A *rectangle-DAG* solving $\mathsf{Search}(F)$ is a DAG Π of out-degree at most 2, with a distinguished root node $r \in \Pi$ and a rectangle $R_v \subseteq X \times Y$ associated to every $v \in \Pi$, such that

1. the root rectangle is full, i.e., $R_r = X \times Y$;
2. the rectangle at a non-leaf node v is covered by its out-neighbors v_1, v_2, i.e., $R_v \subseteq R_{v_1} \cup R_{v_2}$; and
3. the rectangle R_ℓ at every leaf ℓ is monochromatic for some clause C.

The *size* of a rectangle-DAG is the number of nodes $|\Pi|$ in it.

Sunflowers. We will make use of *robust sunflowers* for constructing our approximators.

Definition 2.1. For a family $\mathcal{F}$ of subsets of $[n]$, we call $K = \bigcap_{F \in \mathcal{F}} F$ the *core of* $\mathcal{F}$. Given $x \in \{0,1\}^n$, we write $F \subseteq x$ if for all $i \in F$, $x_i = 1$, and consider F as an element of $\{0,1\}^n$ in the analogous way. We say that $\mathcal{F}$ is an *ε-robust sunflower* if

$$\Pr_{\mathbf{Y} \sim \{0,1\}^n}[\exists F \in \mathcal{F} \colon F \subseteq \mathbf{Y} \cup K] \geq 1 - \varepsilon.$$

Below, we state the currently best known bounds for the robust sunflower lemma.

Lemma 2.2 (Robust sunflower lemma, [BCW21]). There is a constant $c > 0$ such that every family $\mathcal{F}$ of sets of size k, if $|\mathcal{F}| \geq (c \log(k/\varepsilon))^k$, then $\mathcal{F}$ contains an ε-robust sunflower.

3 Expansion Properties of Random Bipartite Graphs

We consider the following two notions of expansion in bipartite graphs.

Definition 3.1. A bipartite graph $G = (L, R, E)$ is called an (r, Δ, c)-*expander* if all left vertices $v \in L$ have degree at most Δ, and for every set $S \subseteq L$ of size at most $|S| \leq r$, its neighbourhood satisfies $|N(S)| \geq c|S|$.

The graph G is an (r, Δ, c)-*boundary expander* if all left vertices $v \in L$ have degree at most Δ, and for every set $S \subseteq L$ of size at most $|S| \leq r$, its unique neighbourhood satisfies $|\partial(S)| \geq c|S|$.

In particular, we will care about the expansion properties of the following distribution on bipartite graphs. Consider bipartite graphs on disjoint vertex sets L, and R, with $|L| = m$ and $|R| = n$. The distribution $\mathfrak{G}(m, n, k)$, samples, for every $v \in L$, k uniformly random neighbors among R. Graphs sampled from $\mathfrak{G}(m, n, k)$ have good expansion:

Lemma 3.2 (Theorem A.2 in [Sok24]). For constants $\alpha, d > 0$, let $\Delta = d \log n$, $m \leq \alpha n 2^{\Delta}$. Then, for any constant $\eta > 0$, there is a constant $\kappa > 0$ such that $G \sim \mathfrak{G}(m, n, k)$ is a $(\kappa n/\Delta, \Delta, (1-\eta)\Delta)$-expander with high probability.

Actually, good expansion also implies decent *boundary* expansion, as we record next. The lemma follows from a simple counting argument together with the lemma above.

Lemma 3.3. For constants $\alpha, d > 0$, let $\Delta = d \log n$, $m \leq \alpha n 2^{\Delta}$. Then, for any constant $\eta > 0$, there is a constant $\kappa > 0$ such that $G \sim \mathfrak{G}(m, n, k)$ is a $(\kappa n/\Delta, \Delta, (1-2\eta)\Delta)$-boundary expander with high probability.

Finally, let us see how these statements apply to random CNF formulas.

Definition 3.4. We let $\mathcal{F}(m, n, k)$ denote the distribution on k-CNF formulas over n variables obtained by sampling m clauses uniformly at random (with repetition).

It was shown by Chvátal and Szemerédi [CS88] that $F \sim \mathcal{F}(m, n, k)$ is unsatisfiable with high probability if $m \geq \ln 2 \cdot 2^k n$ for all $k \geq 3$.

With every CNF formula F, we associate a *dependency graph* $G_F = (U, V, E)$ as follows: every vertex in U corresponds to a clause in F; every vertex in V corresponds to a variable; and for $u \in U$, $v \in V$, the edge $(u, v) \in E$ if and only if the variable corresponding to v appears in the clause corresponding to u. Using the lemmas above, one can deduce the well-known fact the dependency graph of a random CNF is an (boundary) expander.

4 An Intermediate Instance Towards Pseudorandom Generator Tautologies

We define the following constraint satisfaction problem, which we view as an intermediate step between random CNF formulas and the inner-product NW generator. We denote by $\mathcal{P}(m, n, k)$ the disribution over sets random parity constraints, namely it samples m uniform random parity constraints over n variables of width k.

Definition 4.1. The distribution $(\mathcal{F} \otimes \mathcal{P})(m, n, \Delta)$ samples m constraints on n Boolean variables. The variables are split into two sets $V_X \sqcup V_Y$ of equal size $n/2$. For every constraint C_i, we sample a uniformly random clause C_i^X of width Δ among the variable set V_X, and a uniformly random parity constraint P_i^Y of width Δ among the variables V_Y. We then define the ith constraint as $C_i := C_i^X \vee P_i^Y$.

By [CS88], the set of constraints sampled from $(\mathcal{F} \otimes \mathcal{P})(m, n, k)$ is unsatisfiable if $m \geq \ln 2 \cdot 2^k n$ for all $k \geq 3$. Denote by L the set of constraints. Let $G_\varphi = (L, V_X \cup V_Y, E)$ be the dependency graph defined in the same way as for random CNF formulas. It holds that Lemma 3.2 and Lemma 3.3 apply to both $G_\varphi[L \cup V_X]$ and $G_\varphi[L \cup V_Y]$ by construction.

Denote by X the assignments to V_X, and by Y the assignments to V_Y. Given an instance $\varphi \sim (\mathcal{F} \otimes \mathcal{P})(m, n, k)$, we define the falsified constraint search problem $\mathrm{Search}(\varphi) \subseteq (X \times Y) \times \varphi$ such that $(x, y, C) \in \mathrm{Search}(\varphi)$ if and only if the assignment $x \cup y$ falsifies the constraint C. We prove the following lower bound for rectangle-DAGs solving $\mathrm{Search}(\varphi)$.

Theorem 4.2. Let $\alpha, d > 0$, be constants and $\Delta = d \log n$, $m = \alpha n 2^{\Delta}$. For $\varphi \sim (F \otimes P)(m, n, \Delta)$, a rectangle-DAG Π solving $\mathrm{Search}(\varphi)$ has length $|\Pi|$ at least $2^{\Omega(n)}$ with high probability.

It is clear that searching for a falsified constraint in $\varphi \sim (\mathcal{F} \otimes \mathcal{P})(m, n, k)$ is easier than searching for a falsified clause in $F \sim \mathcal{F}(m, n, k)$, since the parity part of the constraints in φ have more falsifying assignments than clauses.

5 Lower Bound for Intermediate Instance

The general plan for the proof of Theorem 4.2 is to adopt Razborov's method of approximations [Raz85] directly to rectangle-DAGs. That is, we define indicators and approximators directly for the rectangles in the protocol.

5.1 Pruning

The goal of this pruning step is to remove all weakening nodes, i.e., nodes v whose rectangles R_v are entirely covered by the rectangle at one of the out-neighbors of v. To achive this, we traverse a rectangle-DAG Π' solving $\mathrm{Search}(\varphi)$ in topological order. Whenever, at a node v, for one of its children u, it holds $R_v \subseteq R_u$, we delete v and insert an edge to u form all the in-neighbors of v. The resulting rectangle-DAG Π clearly still solves S, and now has the additional property that no node in Π is completely covered by one of its out-neighbors, that is, for $v \in \Pi$ with out-neighbors v_1, v_2, it holds $R_v \setminus R_{v_1} \neq \emptyset$ and $R_v \setminus R_{v_2} \neq \emptyset$.

5.2 Approximator in general terms

We start by defining our approximators. Loosely speaking, these will be an OR of *indicators*, which in turn behave like an AND of several monochromatic rectangles.

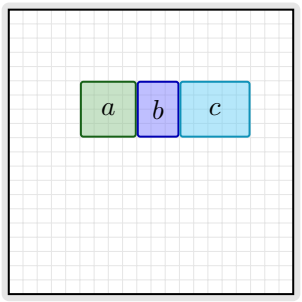
Let S be a set of constraints of φ, and fix a C-monochromatic rectangle $R_C \subseteq X \times Y$ for every $C \in S$. We say that a rectangle $I_S := S_X \times S_Y \subseteq X \times Y$ is an *indicator* iff $S_X \subseteq R_C$ for every $C \in S$. We say that rectangle $A_{\mathcal{J}}$ is an *approximator* based on collection of indicators $\mathcal{J}$ iff $A_{\mathcal{J}} \subseteq \bigcup_{I \in \mathcal{J}} I$. See Figure 1 for an illustration. It may be useful to keep the following intuition in mind: An indicator can implement the logical AND of a collection of monochromatic rectangles, whereas an approximator may behave like the logical OR of a collection of indicators.

Denote by $\mathcal{J}_k \subseteq \mathcal{J}$ the subset of indicators I_S with $|S| = k$. The approximator $A_{\mathcal{J}}$ is called an (ε, s)-*approximator* if

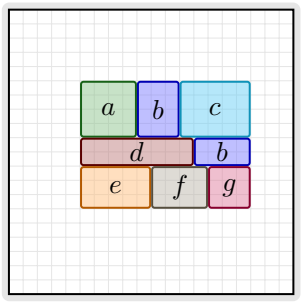
1. for all $k \leq s$, $|\mathcal{J}_k| \leq (c \log(k/\varepsilon))^k$; and
2. for all $k > s$, $\mathcal{J}_k = \emptyset$.

Hereafter, we set $w = \frac{n}{\log n}$ and $\varepsilon = n^{-3w}$. Choose d large enough so that $d > 3/(1 - \eta)$. Additionally, we let $s = \lambda n / \log n < \kappa n/(2\Delta)$ for some constant $\lambda > 0$.

To deduce our lower bound on the size of rectangle-DAGs, we need to show that



(a) An indicator intersecting three monochromatic rectangles.



(b) An approximator built from three indicators. Note that the monochromatic rectangle b appears in the set of constraints of two indicators.

Figure 1: Example of indicator and approximator.

1. an (ε, s)-approximator cannot cover the whole of $X \times Y$; and
2. we can approximate every step in the rectangle-DAG without incurring too much error.

5.3 Approximator Does Not Cover Domain

For item 1, we show the following lemma.

Lemma 5.1. An (ε, s)-approximator A covers at most an $o(1)$-fraction of the inputs, i.e.,

$$\Pr_{\mathbf{x} \in X \times Y}[\mathbf{x} \in A] = o(1).$$

Proof. We will show that $A_{\mathcal{J}} \restriction_X$ covers only an $o(1)$-fraction of X. Let $I_S \in \mathcal{J}$ be an indicator of size $|S| = k$. It holds that at least $(1-\eta)\Delta k$ distinct variables appear in the clauses of S. Thus, the density of $I_S \restriction_X$ in X is at most $2^{-(1-\eta)\Delta k}$. By a union bound over all at most $(c \log(k/\varepsilon))^k$ indicators of size k in $A_{\mathcal{J}}$, we bound the density in X of points covered by the indicators of size k:

$$\frac{(c\log(k/\varepsilon))^k}{2^{(1-\eta)\Delta k}} = \frac{(c\log(k/n^{-3w}))^k}{2^{(1-\eta)d\log(n)k}} \leq \frac{(3(c+1)n)^k}{n^{(1-\eta)dk}} < \frac{(3(c+1))^k}{n^k} = O((\log\log n/n)^k).$$

Summing over all possible sizes of inicators in $A_{\mathcal{J}}$, we get that $A_{\mathcal{J}} \restriction_X$ covers at most a fraction of $s \cdot O(\log\log n/n) = O(\log\log n/\log n) = o(1)$ of X. The lemma follows, as now

$$\Pr_{\mathbf{x} \in X \times Y}[\mathbf{x} \in A] \leq \Pr_{\mathbf{x} \in X \times Y}[\mathbf{x} \restriction_X \in A \restriction_X] = o(1). \qquad \square$$

5.4 Building the Approximator

Towards item 2, we describe how to construct an approximator gate-by-gate. The next claim will be useful to argue that in this process, when building the approximator for a node v, the rectangles at the out-neighbors of v either cover R_v "horizontally" or "vertically".

Claim 5.2. Suppose $R, R_1, R_2 \subseteq X \times Y$ are rectangles such that $R \subseteq R_1 \cup R_2$ and $R \not\subseteq R_1$, $R \not\subseteq R_2$. Then, either

(a) $R \restriction_X \subseteq R_1 \restriction_X \cap R_2 \restriction_X$, or

Algorithm 1 Plucking procedure

Input: $\mathcal{I}$ a set of indicators , $\mathcal{I}_k \subseteq \mathcal{I}$ the subsets of indicators of size k for every $k \in [s]$.
Output: Set of indicators pluck($\mathcal{I}$) satisfying item 1.
for $k = s, s-1, \ldots, 0$ **do**
  while $|\mathcal{I}_k| > (c \log(k/\varepsilon))^k$ **do**
    Find a sunflower $\mathcal{S}'$ among $\mathcal{S}_k = \{S \mid I_S \in \mathcal{I}_k\}$.
    Replace $\mathcal{S}'$ with its core K, i.e., $\mathcal{S}_k \leftarrow (\mathcal{S}_k \setminus \mathcal{S}') \cup \{K\}$.
return $\mathcal{I}$

(b) $R\restriction_Y \subseteq R_1\restriction_Y \cap R_2\restriction_Y$.

Proof. Suppose the claim is false, and let $x \in R\restriction_X \setminus R_1\restriction_X \cap R_2\restriction_X$, and $y \in R\restriction_Y \setminus R_1\restriction_Y \cap R_2\restriction_Y$. Since $(x, y) \in R$, the covering property implies that (x, y) is in one of R_1 or R_2, so suppose it is in R_1 without loss of generality. This means that $x \notin R_2\restriction_X$ and $y \notin R_2\restriction_Y$. Since $R \not\subseteq R_1$, let (x', y') be any point in $R \setminus R_1$, and hence either $x' \notin R_1\restriction_X$ or $y' \notin R_1\restriction_Y$. By rectangularity, both $(x, y'), (x', y) \in R$. If $x' \notin R_1\restriction_X$, then $(x', y) \notin R_1 \cup R_2$, and if $y' \notin R_1\restriction_Y$, then $(x, y') \notin R_1 \cup R_2$. But then one of these two points is not covered by $R_1 \cup R_2$, a contradiction. □

The process of building the approximator for Π is bottom-up, i.e., we traverse Π in reverse topological order and inductively build an approximator for the root, by building an approximator A_v for every node on the way. We will also keep track of *errors*. We say that $x \in X$ is an *error row at* v if $x \in R_v\restriction_X \setminus A_v\restriction_X$ and that $y \in Y$ is an *error column at* v if $y \in R_v\restriction_Y \setminus A_v\restriction_Y$. That is to say, we are measuring the error where the approximator A_v *does not* cover some part of R_v, but not vice-versa. We denote the sets of error rows and columns at v as Err_v^X and Err_v^Y, respectively. We also want to be able to assign an error row or column to the particular node in the approximation process where it was incurred. For this purpose, we define *accumulated error rows at* v as $\mathrm{AccErr}_v^X = \bigcup_{u \in \mathrm{desc}(v)} \mathrm{Err}_u^X$ and the *accumulated error colums at* v as $\mathrm{AccErr}_v^Y = \bigcup_{u \in \mathrm{desc}(v)} \mathrm{Err}_u^Y$. Now, to achieve item 2, we simply need to bound $\mathrm{Err}_v^X \setminus \mathrm{AccErr}_v^X$ and $\mathrm{Err}_v^Y \setminus \mathrm{AccErr}_v^Y$.

At a leaf $\ell \in \Pi$, there is a monochromatic rectangles, which we simply use as our approximator (it has exactly one indicator, which is of size one), so $\mathrm{Err}_\ell^X = \mathrm{Err}_\ell^Y = \emptyset$.

Assume that we reach a node $v \in \Pi$, labelled by the rectangle $R \subseteq X \times Y$ with children v_1, v_2, labelled by the rectangles R_1 and R_2. By the induction hypothesis, we have already built the approximators $A_{\mathcal{I}_1}$ and $A_{\mathcal{I}_2}$ for R_1 and R_2. By Claim 5.2, there are two cases.

We consider item (a) first. Let $\mathcal{S}_1 = \{S \mid I_S \in \mathcal{I}_1\}$ and $\mathcal{S}_2 = \{S \mid I_S \in \mathcal{I}_2\}$. We define $\mathcal{I} = \{I_{S_1 \cup S_2} \mid S_1 \in \mathcal{S}_1, S_2 \in \mathcal{S}_2\}$ and denote $\mathcal{A}_v = A_{\mathcal{I}}$. Note that so far, we have not introduced any new error columns: it holds that $A_{\mathcal{I}_1} \cup A_{\mathcal{I}_2} \cup (\mathrm{AccErr}_v^X \times \mathrm{AccErr}_v^Y) \supseteq R_v$, and if $y \in (A_{\mathcal{I}_1} \cup A_{\mathcal{I}_2})\restriction_Y$, then $y \in \mathcal{A}_v\restriction_Y$. We have also not introduced any new error rows by item (a). First, apply the plucking procedure in 1 to $\mathcal{I}$. Then remove all indicators of size larger than s, i.e., $I_S \in \mathcal{I}$, $|S| > s$, from pluck($\mathcal{I}$) , and call the resulting set $\mathcal{I}' \subseteq \mathrm{pluck}(\mathcal{I})$. We then use $A_v = A_{\mathcal{I}'}$ as approximator for v.

For item (b), let $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2$, and denote $\mathcal{A}_v = A_{\mathcal{I}}$. Note that so far, we have not introduced any new error rows: it holds $A_{\mathcal{I}_1} \cup A_{\mathcal{I}_2} \cup (\mathrm{AccErr}_v^X \times \mathrm{AccErr}_v^Y) \supseteq R_v$, and if $x \in (A_{\mathcal{I}_1} \cup A_{\mathcal{I}_2})\restriction_X$, then $x \in \mathcal{A}_v\restriction_X$. We have also not introduced any new error columns by item (b). Then apply the plucking procedure in 1. The approximator for v is $A_v = A_{\mathrm{pluck}(\mathcal{I})}$.

In both cases, new error columns were only introduced via plucking, and error new rows were only introduced via the removal of large indicators.

5.5 Bounding the Approximation Error

Next, we bound the approximation error introduced by the procedure above at a given node.

Lemma 5.3. At every node $v \in \Pi$, building the approximator introduces at most a $2^{-\Omega(n)}$-fraction of points as errors, i.e.,

$$\Pr_{\mathbf{x} \in X \times Y}\left[\mathbf{x} \in \left(\mathrm{Err}_v^X \setminus \mathrm{AccErr}_v^X\right) \times \left(\mathrm{Err}_v^Y \setminus \mathrm{AccErr}_v^Y\right)\right] = 2^{-\Omega(n)}.$$

To prove this lemma, we need a bound on the density of the errors introduced by the plucking procedure:

Claim 5.4. The error incurred by sunflower plucking is bounded by 2ε.

We defer the proof of this claim until after the proof of Lemma 5.3.

Proof of Lemma 5.3. We can bound the number of indicators in any approximator $A = A_{\mathcal{J}}$ as follows:

$$|\mathcal{J}| \leq \sum_{k=1}^{s} (c \log(k/\varepsilon))^k \leq s \cdot c^s \cdot (3w \log n)^s = s \cdot 3^s c^s n^s \leq n^w,$$

where we used that $\kappa/2d < 1/5$ in the last step.

In case (a), there are at most $|\mathcal{J}|^2 \leq n^{2w}$ pluckings, whereas in case (b), there no more than $2|\mathcal{J}| \leq 2n^w$. Applying Claim 5.4, every plucking incurs ε-density of errors, and by a union bound, we have that this leads to $n^{2w} \cdot 2n^{-3w} = n^{-w} = 2^{-n}$ error density.

In case (a), we also remove the indicators of size larger than s. We bound the error density as follows:

$$\begin{aligned}
\sum_{k=s+1}^{2s} (c \log(k/\varepsilon))^k \cdot 2^{-(1-\eta)\Delta k} &\leq s \cdot (c \log(2s/\varepsilon))^{2s} \cdot 2^{-(1-\eta)\Delta s} \\
&= s \cdot (c \log(2s/n^{-3w}))^{2s} \cdot n^{-(1-\eta)ds} \\
&\leq s \cdot \frac{(3(c+1)n)^{2s}}{n^{3s}} \\
&\leq n^{-.9s} = 2^{-\Omega(n)},
\end{aligned}$$

where we used the definition of s in the second inequality. □

Proof of Claim 5.4. Let $\mathcal{S}$ be the robust sunflower found by Algorithm 1, K its core and $U = \bigcup_{S \in \mathcal{S}} S$. Note that if a row corresponding to an X-input intersected the approximator before plucking, the same holds after plucking. Therefore, to bound the density of the errors it is enough to do so for Y. Y-inputs that contribute to the error are those which intersect a rectangle in each petal, but do not intersect the core. For a rectangle $s \in U$, denote its corresponding constraint by $C_s = C_s^X \vee P_s^Y$.

Let us say that a set of constraints is *covered* by an input if all constraints in the set are satisfied by this input. Then, we need to bound the probability that an input covers the core, but does not cover any petal. Let $\mathcal{D}$ be the distribution on $\mathbf{Y} \in \{0,1\}^U$ defined as follows.

- Sample a uniform $\mathbf{x} \in \{0,1\}^{N(U)}$ and set $\mathbf{Y}_s = P_s^Y(\mathbf{x})$.
- Output $\mathbf{Y}$.

It holds that:

$$\begin{aligned}
&\Pr_{\mathbf{Y} \sim \mathcal{D}}[\exists S \in \mathcal{S} \colon S \subseteq \mathbf{Y} \cup K] && \text{(Claim 5.5)} \\
&\geq \Pr_{\mathbf{Y} \sim \{0,1\}^U}[\exists S \in \mathcal{S} \colon S \subseteq \mathbf{Y} \cup K] - \varepsilon && \text{(Robust sunflower properties)} \\
&\geq 1 - 2\varepsilon,
\end{aligned}$$

from which the upper bound on the error density follows. □

We prove the missing claim below.

Claim 5.5.

$$\Pr_{\mathbf{Y}\sim\mathcal{D}}[\exists S\in\mathcal{S}\colon S\subseteq\mathbf{Y}\cup K]\geq\Pr_{\mathbf{Y}\sim\{0,1\}^U}[\exists S\in\mathcal{S}\colon S\subseteq\mathbf{Y}\cup K]-\varepsilon.$$

To prove this claim, we need the following two statements.

Theorem 5.6 (Braverman's theorem, [Bra08]). Let f be a boolean function computed by a size m and depth d circuit. Let $s\geq\log m, r\geq(s\log m)^{\Omega(d^2)}$ be any parameters and μ be an r-wise independent distribution on $\{0,1\}^n$. Then

$$|\mathop{\mathbb{E}}_{\mathbf{x}\sim\mu}[f(\mathbf{x})]-\mathop{\mathbb{E}}_{\mathbf{x}\sim\{0,1\}^n}[f(\mathbf{x})]|\leq\epsilon,$$

where $\epsilon=2^{-\Omega(s)}m$.

Claim 5.7. $\mathcal{D}$ is r-wise independent.

Proof of Claim 5.5. We use the fact that sunflowers are essentially equivalent to monotone DNFs. For every element $s\in U$, consider a formal variable z_s and let $t_S:=\bigwedge_{s\in S}z_s$ be the term that naturally corresponds to a petal $S\in\mathcal{S}$. Observe that

$$\Pr_{\mathbf{Y}\sim\mathcal{D}}[\exists S\in\mathcal{S}\colon S\subseteq\mathbf{Y}\cup K]=\Pr_{\mathbf{Z}\sim\mathcal{D}}\left[\bigvee_{S\in\mathcal{S}}t_S|_{\mathbf{Z}}=1\right]\text{ and}$$

$$\Pr_{\mathbf{Y}\sim\{0,1\}^U}[\exists S\in\mathcal{S}\colon S\subseteq\mathbf{Y}\cup K]=\Pr_{\mathbf{Z}\sim\{0,1\}^U}\left[\bigvee_{S\in\mathcal{S}}t_S|_{\mathbf{Z}}=1\right].$$

Since $\mathcal{D}$ is an r-wise independent distribution by Claim 5.7, we apply Braverman's theorem [Bra08] with parameters $m=2^{n^\gamma}, s=n^{2\gamma}, d=2$ for small enough constant γ and the claim follows. □

Next, we supply the proof of Claim 5.7. It relies on the following statement about bipartite boundary expanders.

Lemma 5.8 ([GMT09; SS25]). Let $G=(L,R,E)$ be (r,Δ,c)-boundary expander and $S\subseteq L$ be a set of vertices with $|S|\leq r$. Then there exists an enumeration $S=\{s_1,\dots,s_{|S|}\}$ and a partition $\bigsqcup R_i=N(S)$ such that:

- $R_i=N(s_i)\setminus\left(\bigcup_{j=1}^{i-1}N(s_j)\right)$;
- $|\partial(s_i)\cap R_i|\geq c$.

Proof of Claim 5.7. Let S the a set of parity constraints, $|S|\leq r$. We use Lemma 5.8 to obtain an enumeration $S=\{s_1,\dots,s_{|S|}\}$ of the vertices with an associated partition $\bigsqcup R_i=N(S)$. Let $\alpha\in\{0,1\}^S$ be any bit string. We prove that $\Pr_{\mathbf{x}\sim\{0,1\}^n}[S(\mathbf{x})=\alpha]=2^{-|S|}$.

The proof is by induction. The base case $S=\emptyset$ is trivial. For the induction step, we apply the hypothesis for the set $S':=S\setminus\{s_{|S|}\}$ and $\alpha':=\alpha_{S'}$. It holds that $\Pr_{\mathbf{x}\sim\{0,1\}^n}[S'(\mathbf{x})=\alpha']=2^{-|S'|}=2^{-|S|+1}$. Moreover, $\Pr_{\mathbf{x}\sim\{0,1\}^n}[s_{|S|}(\mathbf{x})=\alpha_{|S|}\mid S'(\mathbf{x})=\alpha']=\frac{1}{2}$, since $s_{|S|}$ has at least $(1-2\eta)\Delta$ unique neighbours in $R_{|S|}$. □

5.6 Proof of Theorem 4.2

Let r denote the root of of a rectangle-DAG solving Search(φ). Denote by A_r the (s, ε)-approximator constructed as above. The rectangle at the root is $R_r = X \times Y$, so by Lemma 5.1,

$$\Pr_{\mathbf{x} \in X \times Y}[\mathbf{x} \in R_r \setminus A_r] \geq 1 - o(1). \tag{1}$$

We have that

$$R_r \setminus A_r \subseteq \mathrm{Err}_r^X \times \mathrm{Err}_r^Y = \bigcup_{v \in \Pi} (\mathrm{Err}_v^X \setminus \mathrm{AccErr}_v^X) \times (\mathrm{Err}_v^Y \setminus \mathrm{AccErr}_v^Y),$$

so by a union bound and Lemma 5.3, it holds

$$\Pr_{\mathbf{x} \in X \times Y}[\mathbf{x} \in R_r \setminus A_r] \leq \sum_{v \in \Pi} 2^{-\Omega(n)}.$$

Together with (1), this implies that $|\Pi| \geq 2^{\Omega(n)}$.

References

[ABE02] Albert Atserias, Maria Luisa Bonet, and Juan Luis Esteban. "Lower Bounds for the Weak Pigeonhole Principle and Random Formulas beyond Resolution." In: *Information and Computation* 176.2 (2002), pp. 136–152.

[ABRW04] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. "Pseudorandom Generators in Propositional Proof Complexity." In: *SIAM Journal on Computing* 34.1 (2004). Preliminary version in *FOCS '00*, pp. 67–88.

[ALWZ21] Ryan Alweiss, Shachar Lovett, Kewen Wu, and Jiapeng Zhang. "Improved bounds for the sunflower lemma." In: *Annals of Mathematics* 194.3 (Nov. 2021). Preliminary version in *STOC '20*, pp. 795–815.

[BCW21] Tolson Bell, Suchakree Chueluecha, and Lutz Warnke. "Note on sunflowers." In: *Discrete Mathematics* 344.7 (2021), p. 112367. ISSN: 0012-365X. DOI: https://doi.org/10.1016/j.disc.2021.112367.

[BI99] Eli Ben-Sasson and Russell Impagliazzo. "Random CNF's are Hard for the Polynomial Calculus." In: *40th Annual Symposium on Foundations of Computer Science, FOCS 1999, New York, NY, USA, October 17-18, 1999*. IEEE Computer Society, 1999, pp. 415–421. DOI: 10.1109/SFFCS.1999.814613. URL: https://doi.org/10.1109/SFFCS.1999.814613.

[BP96] Paul Beame and Toniann Pitassi. "Simplified and Improved Resolution Lower Bounds." In: *37th Annual Symposium on Foundations of Computer Science, FOCS 1996, Burlington, Vermont, USA, 14-16 October, 1996*. IEEE Computer Society, 1996, pp. 274–282. DOI: 10.1109/SFCS.1996.548486. URL: https://doi.org/10.1109/SFCS.1996.548486.

[BPR97] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. "Lower Bounds for Cutting Planes Proofs with Small Coefficients." In: *J. Symb. Log.* 62.3 (1997), pp. 708–728. DOI: 10.2307/2275569. URL: https://doi.org/10.2307/2275569.

[Bra08] Mark Braverman. "Polylogarithmic independence fools AC0 circuits." In: *J. ACM* 57.5 (June 2008). ISSN: 0004-5411. DOI: 10.1145/1754399.1754401.

[CFKMP19] Arkadev Chattopadhyay, Yuval Filmus, Sajin Koroth, Or Meir, and Toniann Pitassi. "Query-To-Communication Lifting for BPP Using Inner Product." In: *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Ed. by Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi. Vol. 132. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019, 35:1–35:15. ISBN: 978-3-95977-109-2. DOI: 10.4230/LIPIcs.ICALP.2019.35. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2019.35.

[CS88] Vašek Chvátal and Endre Szemerédi. "Many Hard Examples for Resolution." In: *Journal of the ACM* 35.4 (Oct. 1988), pp. 759–768.

[DT20] Daniel Dadush and Samarth Tiwari. "On the Complexity of Branching Proofs." In: *35th Computational Complexity Conference (CCC 2020)*. Ed. by Shubhangi Saraf. Vol. 169. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020, 34:1–34:35. ISBN: 978-3-95977-156-6. DOI: 10.4230/LIPIcs.CCC.2020.34. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CCC.2020.34.

[FPPR22] Noah Fleming, Denis Pankratov, Toniann Pitassi, and Robert Robere. "Random $\Theta(\log n)$-CNFs are Hard for Cutting Planes." In: *Journal of the ACM* 69.3 (2022). Preliminary version in *FOCS '17*, 19:1–19:32. DOI: 10.1145/3486680.

[GGKS20] Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. "Monotone Circuit Lower Bounds from Resolution." In: *Theory of Computing* 16.13 (2020), pp. 1–30. DOI: 10.4086/toc.2020.v016a013. URL: https://theoryofcomputing.org/articles/v016a013.

[GGRT26] Michal Garlík, Svyatoslav Gryaznov, Hanlin Ren, and Iddo Tzameret. "The Weak Rank Principle: Lower Bounds and Applications." In: *Proceedings of the 58th Annual ACM Symposium on Theory of Computing, STOC 2026, Salt Lake City, UT, USA, June 22-26, 2026*. Ed. by Aditya Bhaskara and Artur Czumaj. ACM, 2026, pp. 138–149. DOI: 10.1145/3798129.3800735. URL: https://doi.org/10.1145/3798129.3800735.

[GMT09] Konstantinos Georgiou, Avner Magen, and Madhur Tulsiani. "Optimal Sherali-Adams Gaps from Pairwise Independence." In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 12th International Workshop, APPROX 2009, and 13th International Workshop, RANDOM 2009, Berkeley, CA, USA, August 21-23, 2009. Proceedings*. Vol. 5687. Lecture Notes in Computer Science. Springer, 2009, pp. 125–139. DOI: 10.1007/978-3-642-03685-9_10.

[Hak95] Armin Haken. "Counting Bottlenecks to Show Monotone P <=> NP." In: *36th Annual Symposium on Foundations of Computer Science, FOCS 1995, Milwaukee, Wisconsin, USA, 23-25 October 1995*. IEEE Computer Society, 1995, pp. 36–40. DOI: 10.1109/SFCS.1995.492460. URL: https://doi.org/10.1109/SFCS.1995.492460.

[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)*. 2017, pp. 121–131. DOI: 10.1109/FOCS.2017.20.

[Juk97] Stasys Jukna. "Finite Limits and Monotone Computations: The Lower Bounds Criterion." In: *Proceedings of the Twelfth Annual IEEE Conference on Computational Complexity, Ulm, Germany, June 24-27, 1997*. IEEE Computer Society, 1997, pp. 302–313. DOI: 10.1109/CCC.1997.612325. URL: https://doi.org/10.1109/CCC.1997.612325.

[Kra01] Jan Krajíček. "On the weak pigeonhole principle." In: *Fundamenta Mathematicae* 170.1-3 (2001), pp. 123–140.

[Kra97] Jan Krajíček. "Lower Bounds for a Proof System with an Exponential Speed-up over Constant-Depth Frege Systems and over Polynomial Calculus." In: *Mathematical Foundations of Computer Science 1997, 22nd International Symposium, MFCS'97*. 1997, pp. 85–90.

[LMMPZ22] Shachar Lovett, Raghu Meka, Ian Mertz, Toniann Pitassi, and Jiapeng Zhang. "Lifting with Sunflowers." In: *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Ed. by Mark Braverman. Vol. 215. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 104:1–104:24. ISBN: 978-3-95977-217-4. DOI: 10.4230/LIPIcs.ITCS.2022.104. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2022.104.

[Pud97] Pavel Pudlák. "Lower Bounds for Resolution and Cutting Plane Proofs and Monotone Computations." In: *J. Symb. Log.* 62.3 (1997), pp. 981–998. DOI: 10.2307/2275583. URL: https://doi.org/10.2307/2275583.

[Raz15] Alexander A. Razborov. "Pseudorandom Generators Hard for k-DNF Resolution and Polynomial Calculus Resolution." In: *Annals of Mathematics* 181.2 (Mar. 2015), pp. 415–472.

[Raz85] Alexander A. Razborov. "Lower bounds for the monotone complexity of some Boolean functions." In: *Doklady Akademii Nauk SSSR* 281.4 (1985). English translation in Soviet Math. Doklady, 31:354-357, 1985, pp. 798–801.

[Ros10] Benjamin Rossman. "The Monotone Complexity of k-clique on Random Graphs." In: *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, Las Vegas, Nevada, USA, October 23-26, 2010*. IEEE Computer Society, 2010, pp. 193–201. DOI: 10.1109/FOCS.2010.26. URL: https://doi.org/10.1109/FOCS.2010.26.

[SBI04] Nathan Segerlind, Samuel R. Buss, and Russell Impagliazzo. "A Switching Lemma for Small Restrictions and Lower Bounds for k-DNF Resolution." In: *SIAM Journal on Computing* 33.5 (2004), pp. 1171–1200.

[Sok17] Dmitry Sokolov. "Dag-Like Communication and Its Applications." In: *Proceedings of the 12th International Computer Science Symposium in Russia (CSR '17)*. Vol. 10304. Lecture Notes in Computer Science. Springer, 2017, pp. 294–307. DOI: 10.1007/978-3-319-58747-9_26.

[Sok22] Dmitry Sokolov. "Pseudorandom Generators, Resolution and Heavy Width." In: *Proceedings of the 37th Computational Complexity Conference (CCC '22)*. Vol. 234. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 15:1–15:22. ISBN: 978-3-95977-241-9. DOI: 10.4230/LIPIcs.CCC.2022.15.

[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)*. 2024, pp. 2008–2015. DOI: 10.1145/3618260.3649636.

[SS25] Anastasia Sofronova and Dmitry Sokolov. "A Lower Bound for k-DNF Resolution on Random CNF Formulas via Expansion." In: *Proceedings of the 40th Computational Complexity Conference (CCC '25)*. Vol. 339. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025, 32:1–32:27. DOI: 10.4230/LIPICS.CCC.2025.32.