THEORY OF COMPUTING
www.theoryofcomputing.org

A Simplified Proof for the Monomial Space Complexity of Random 4-CNF Formulas in Polynomial Calculus

Massimo Lauria[*] Mladen Mikša Jakob Nordström[†] Marc Vinyals[‡]

December 29, 2025

Abstract. We present a simpler, more explicit, proof of the asymptotically optimal linear lower bound on the monomial space required to refute random 4-CNF formulas in polynomial calculus as shown by [Bonacina and Galesi '15]. We hope that this exposition can help make the result more accessible, and perhaps also provide useful insights for other open problems concerning polynomial calculus space.

1 Introduction

Proof complexity studies how hard it is to refute unsatisfiable formulas in conjunctive normal form, i.e., the complement of the SAT problem. Clearly, satisfiable formulas have concise witnesses, as per the definition of NP, but it is less clear how to efficiently certify unsatisfiability. This is just another way of

[*] Partially funded by European Research Council grant ERC-2014-CoG 648276 (AUTAR).

[†] Supported by the Swedish Research Council grant 2016-00782 and the Independent Research Fund Denmark grant 9040-00389B.

[‡] This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 802020-ERC-HARMONIC.

ACM Classification: F.2.2, F.4.1

AMS Classification: 03F20

Key words and phrases: proof complexity, polynomial calculus, space complexity, monomial space, random CNF formulas

asking how coNP is related to NP, and historically, therefore, the main focus in proof complexity has been on proof size. In the late 1990s, however, work began on studying also the space complexity of proofs. The initial motivation for this came from SAT solving, where memory management is a major concern, but space complexity is also a fundamental measure in its own right in computational complexity theory, and the study of space in proof complexity has revealed many connections and questions of intrinsic interest.

To give an informal description of the model, suppose that we have a blackboard on which the proof is presented. The size of a proof is then the number of symbols written on the board. But suppose that we can erase from the board, and that we have the input formula in read-only memory and do not need to have it at all times on the board. Then the space of a proof is how large the board needs to be for a self-contained presentation of the proof, where inference rules can only be applied to formulas currently on the board, but where partial results can be erased either when no longer needed or when space is tight (and we are willing to do the extra work of rederiving the same results later when needed again).

In the interest of brevity, we will focus the rest of this introduction on the space measure in proof complexity, referring the reader to, e.g., the book [20] or the survey chapter [10] for more information about proof complexity in general.

1.1 Previous Work on Proof Space Complexity

A formal definition of proof space complexity was first given for the resolution proof system in [13], and this definition was generalized to polynomial calculus[1] and other proof systems in [1]. In resolution new clauses implied by any pair of previous clauses can be derived until the empty clause is obtained, showing that the given formula is unsatisfiable. In polynomial calculus clauses are translated to multilinear polynomials over some field in formal variables x and $\overline{x}$ for all literals occurring in the formula, and one can then take linear combinations and multiply by variables to derive contradiction, i.e., by showing that the polynomial 1 lies in the ideal generated by the polynomials corresponding to the clauses of the CNF formula. Additional polynomials $x^2 - x$ and $x + \overline{x} - 1$ enforce that only Boolean assignments are considered and that the meaning of negation is respected. To date, most works on proof space complexity have studied *clause space* in resolution and *monomial space* in polynomial calculus, measuring the maximal number of clauses/monomials needed simultaneously in a self-contained presentation of a proof that a formula is unsatisfiable.[2]

The focus of this paper is polynomial calculus, but to get a sense for how to prove space lower bounds it is instructive to first look at the simpler case of resolution. Almost all lower bounds on resolution clause space go as follows: Suppose that we want to prove that refuting a CNF formula F requires space at least s. Then, writing $\mathbb{C}_t$ to denote the clauses on the blackboard at time t, we can equivalently show that no derivation $\pi = (\mathbb{C}_1 = \emptyset, \mathbb{C}_2, \ldots, \mathbb{C}_\tau)$ in space less than s can derive contradiction. We do so by inductively constructing partial truth value assignments α_t that assign at most $|\mathbb{C}_t|$ variables and that satisfy all the clauses in $\mathbb{C}_t$ (meaning, in particular, that contradiction has not been derived). The case analysis when we go from $\mathbb{C}_t$ to $\mathbb{C}_{t+1}$ and need to build α_{t+1} from α_t, is as follows:

[1] In this introductory overview we do not distinguish between *polynomial calculus (PC)* [12] and the slightly more general proof system *polynomial calculus resolution (PCR)* [1], the latter being more relevant from the point of view of space complexity.

[2] We mention for completeness that there is also a *total space* measure counting the total number of variables in memory with repetitions, which has been studied in [1, 6, 7, 9], but we do not discuss this measure in the current paper.

1. If $\mathbb{C}_{t+1}$ is obtained from $\mathbb{C}_t$ by inferring a new clause C from clauses on the board, then α_t already satisfies C (by the soundness of the proof system) and we can set $\alpha_{t+1} = \alpha_t$.

2. If $\mathbb{C}_{t+1}$ is obtained from $\mathbb{C}_t$ by erasing clauses from the board, then α_t also satisfies $\mathbb{C}_{t+1}$, but we need to shrink the assignment to at most $|\mathbb{C}_{t+1}|$ variables. This is not a problem, though, since we can just choose from α_t one variable per clause in $\mathbb{C}_{t+1}$ that is assigned so as to satisfy the clause in question.

3. The tricky case is when a new *axiom clause* C from F is copied to the board. Now we need to argue that if α_t does not already satisfy $\mathbb{C}_{t+1}$ but has small enough domain, then we can find a variable x in C that is not assigned by α_t, and extend α_t to α_{t+1} by setting this variable x so as to satisfy C. In general, there might not exist such an unassigned variable x in C, but if the formula F has certain expansion properties and C is not too large, then one can show that such a variable exists.

In this way optimal lower bounds on clause space were shown in [1, 4, 13] for many formula families commonly studied in proof complexity. With hindsight, all of these bounds turn out to follow from lower bounds on the *width* of refutations, measured as the size of a largest clause in any refutation [2] (see also [15]). A sequence of papers [22, 24, 5] later separated the space and width measures by proving space lower bounds using fairly different techniques based on pebble games [23, 25], that we do not discuss here.

For polynomial calculus the approach above provably does not work, however, as demonstrated by Alekhnovich et al. [1]. What fails is step 2: the derivation may contain small configurations that are satisfiable but not by setting just one variable per monomial, contrary to what happens in resolution. As a toy example, suppose that we have input polynomials $x_1 - 1, x_2 - 1, \ldots, x_n - 1$ that are all copied to the board, after which the polynomial $x_1 x_2 \cdots x_n - 1$ is derived (which is possible, since polynomial calculus can derive any implied polynomial) and the input polynomials are erased. Then the polynomial left on the board contains only two monomials, and yet one needs to assign all n variables to 1 to make sure that the polynomial is satisfied and vanishes. But if we can instead pick *two* new, unassigned variables x, y for each new clause C in step 3, choose a 2-clause $D \subseteq C$ over these variables, and carry out the line of argument above for the set of all assignments that satisfy such clauses D, step 2 can be made to work also for polynomial calculus. This fact, referred to as the *locality lemma* by Alekhnovich et al. [1], is highly nontrivial, although the proof boils down to elementary combinatorics. Such locality lemmas lie at the foundation also of the ensuing polynomial calculus space lower bounds discussed below. Thus, instead of maintaining satisfying partial assignments α_t, which can be viewed as 1-CNF formulas, we construct satisfiable 2-CNF formulas $\mathbb{D}_t$ that "overapproximate" the set of polynomials $\mathbb{P}_t$ currently on the board in the sense that $\mathbb{D}_t \vDash \mathbb{P}_t$.

There are additional technical complications, however, and the arguments in [1] are carried out in multi-valued logic, which limits the technique to formulas with very large clauses (such as pigeonhole principle formulas). Filmus et al. [17] translated the technique to standard 2-valued logic and established space lower bounds for specially tailored 4-CNF formulas, but in order for this to work the implications $\mathbb{D}_t \vDash \mathbb{P}_t$ could be made to hold only for a certain subset of "well-behaved" assignments (and satisfiability of $\mathbb{D}_t$ hence had to be shown with respect to such assignments). Also, both [1] and [17] failed to prove space lower bounds matching the known linear upper bounds (measured in terms of the number of variables of a formula).

In something of a breakthrough result, Bonacina and Galesi [8] presented a unified framework for proving lower bounds on monomial space in polynomial calculus, capturing all the results in [1, 17] and finally establishing optimal, linear lower bounds for randomly sampled 4-CNF formulas. These lower bounds were achieved at the price of further complications in the proofs, however. The main technical contribution of [8] is to define certain abstract combinatorial objects and show that if such objects can be constructed for a family of formulas, then space lower bounds follow. It appears quite hard to obtain from this combinatorial argument any intuition about what is actually going on in the polynomial calculus derivation. Following this, [6] managed to extend [8] to few more interesting cases by further refining the technical details of the framework.

1.2 Our Contribution

We extract from the machinery developed by Bonacina and Galesi [8] an explicit, simple proof of the optimal, linear, lower bound on monomial space for polynomial calculus refutations of randomly sampled 4-CNF formulas. Given a polynomial calculus derivation $\pi = (\mathbb{P}_0 = \emptyset, \mathbb{P}_1, \ldots, \mathbb{P}_\tau)$ in sublinear space, we show that asymptotically almost surely (over the randomly sampled formula) one can construct a sequence of *shadow configurations* $(\mathbb{D}_0 = \emptyset, \mathbb{D}_1, \ldots, \mathbb{D}_\tau)$ with the following properties:

P1. Each $\mathbb{D}_t = Q_t \wedge M_t$ consists of two 2-CNF formulas Q_t and M_t, where distinct pairs of clauses in Q_t do not share any variables, and nor do pairs of clauses in M_t; each variable in each clause in M_t is shared with exactly one clause in Q_t; and each clause in Q_t shares exactly one variable with exactly one clause in M_t. The total number of clauses in $\mathbb{D}_t$ is at most six times the number of monomials in $\mathbb{P}_t$.

P2. Every clause in Q_t is a subclause of some axiom clause in the random CNF formula F that has previously been on the board.

P3. For the clauses in M_t we have no such structural information, but the variables in M_t come from axiom clauses in F that have previously been on the board.

P4. The implication $\mathbb{D}_t \vDash \mathbb{P}_t$ holds over all truth value assignments (well-behaved or not, and in standard 2-valued logic).

It follows from property P1 that each $\mathbb{D}_t$ is satisfiable (first pick a satisfying assignment to M_t, and then assign free variables to satisfy also Q_t), and hence so is $\mathbb{P}_t$ by property P4. Perhaps more interestingly, properties P2 and P3 provide some intuition about what polynomials derived in small space must look like. It is a fairly standard fact that a randomly sampled 4-CNF formula has good expansion properties, i.e., for any small-to-medium-large set of clauses there will be many variables that occur only in a single clause, and property P2 is derived from that fact. What property P3 says is that even as we take linear combinations of the corresponding polynomials and multiply them with other variables, it is not possible to cancel these variables.

The 2-CNF formulas $\mathbb{D}_t$ actually give slightly more precise structural information than this. It follows from the properties above that if a polynomial p with s monomials is derivable in sublinear space from a randomly sampled 4-CNF formula, then we can force p to evaluate to 0 by assigning at most $\mathrm{O}(s)$ variables, notwithstanding the fact that the monomials in p might have high degree. Thus, in particular, polynomials such as $x_1x_2\cdots x_n - 1$ discussed above cannot be derived in small space from a randomly sampled CNF formula (where all of these claims hold asymptotically almost surely). We find this to be quite an intriguing fact.

We want to emphasize that we do not present any additional technical developments of tools in this paper on top of what can be found in [8]. However, by using and simplifying the ideas in [8] for the particular case of random CNF formulas we can obtain a much more transparent proof, which also allows us to derive the kind of structural information about the derived polynomials discussed in the previous paragraph. We hope that this simpler, more explicit, presentation of the result in [8] can make it easier to understand, and can stimulate further research on the many open problems that remain regarding space complexity in algebraic proof systems.

1.3 Outline of This Paper

After giving the necessary preliminaries in Section 2, we present our simplified proof of the lower bound by Bonacina and Galesi [8] in Section 3. We conclude by discussing some open problems in Section 4. The proofs for some more standard technical material are given in Appendix A for completeness.

2 Preliminaries

Let us start by quickly reviewing some basics. A *literal* over a Boolean variable x is either the variable x itself or its negation $\overline{x}$. A *k-clause* $C = a_1 \vee \cdots \vee a_k$ is a disjunction of exactly k literals. A *k-CNF formula* $F = C_1 \wedge \cdots \wedge C_m$ is a conjunction of k-clauses. We think of clauses and CNF formulas as sets, so that order is irrelevant and there are no repetitions. Without loss of generality, we only consider non-trivial clauses that do not contain both a variable and its negation.

Let $\mathbb{F}$ be a field and consider the ring $\mathbb{F}[x,\overline{x},y,\overline{y},\ldots]$ of polynomials, where x and $\overline{x}$ are distinct formal variables intended to encode opposite literals. (Importantly, all truth value assignments below only consider only variables x,y,z, however, and automatically give opposite truth values to opposite literals.) A *monomial* m is a product of variables and a *term* $t = \alpha m$ is a monomial m multiplied by a non-zero field element α. We represent *polynomials* p as sums of terms $p = \sum_i t_i = \sum_i \alpha_i m_i$ over distinct monomials m_i. We identify 0 with true and 1 with false,[3] and sometimes adopt the notation $x^0 = x$ and $x^1 = \overline{x}$ for polarities of literals, so that x^b is the literal that vanishes when $x = b$. This should not be any source of confusion, since in the context of polynomial calculus we can think of all polynomials as being multilinear without any loss of generality (i.e., any variables appear with exponent 1 in any monomial). We can represent a clause $C = \bigvee x_i^{b_i}$ by forming the monomial $m_C = \prod x_i^{b_i}$ that is the product of its literals, and we have that an assignment satisfies a clause C if and only if it makes the corresponding monomial m_C evaluate to zero, i.e., if $x_i = b_i$ for some literal $x_i^{b_i}$ in the clause/monomial. For any CNF formula (or set of clauses) F and any sets of polynomials $\mathbb{P}$ we let $Vars(F)$ and $Vars(\mathbb{P})$ denote the set of variables in F and $\mathbb{P}$, respectively.

Definition 2.1 (Polynomial calculus resolution [1, 12]). A *polynomial calculus resolution (PCR) configuration* $\mathbb{P}$ is a set of polynomials over $\mathbb{F}[x,\overline{x},y,\overline{y},\ldots]$. A *PCR derivation* from a CNF formula F is a

[3] Thinking of 0 as true and 1 as false is the opposite of what is commonly done in theoretical computer science, but this convention makes more sense in the context of algebraic proof systems, where “being satisfied” for a polynomial is the same as “vanishing”, i.e., evaluating to zero.

sequence of configurations $\pi = (\mathbb{P}_0, \ldots, \mathbb{P}_\tau)$ such that $\mathbb{P}_0 = \emptyset$ and for $1 \leq t \leq \tau$ the configuration $\mathbb{P}_t$ is obtained from $\mathbb{P}_{t-1}$ by applying one of the following steps:

Axiom download $\mathbb{P}_t = \mathbb{P}_{t-1} \cup \{p\}$, where p is either the monomial m_C representing a clause $C \in F$, the *Boolean axiom* $x^2 - x$ for some variable x, or the *complementarity axiom* $x + \bar{x} - 1$ for a pair of opposite literals x and $\bar{x}$.

Inference $\mathbb{P}_t = \mathbb{P}_{t-1} \cup \{p\}$, where p is inferred by one of the *inference rules*

- *Linear combination* $\frac{q \quad r}{\alpha q + \beta r}$
- *Multiplication* $\frac{q}{x^b q}$

where q, r are polynomials in $\mathbb{P}_{t-1}$, α, β are elements of the underlying field $\mathbb{F}$, x is a variable of F, and $b \in \{0,1\}$.

Erasure $\mathbb{P}_t = \mathbb{P}_{t-1} \setminus \{p\}$, where p is a polynomial in $\mathbb{P}_{t-1}$.

A *PCR refutation* is a derivation such that the polynomial 1 is in $\mathbb{P}_\tau$.

The *monomial space* of a polynomial $p = \sum_{i=1}^{s} \alpha_i m_i$ is the number of monomials $Sp(p) = s$ in p, and the monomial space of a configuration $\mathbb{P} = \{p_1, \ldots, p_m\}$ is $Sp(\mathbb{P}) = \sum_{j=1}^{m} Sp(p_j)$, i.e., the total number of monomials in $\mathbb{P}$ counted with repetitions.[4] The monomial space of a PCR derivation $\pi = (\mathbb{P}_0, \ldots, \mathbb{P}_\tau)$ is the maximum monomial space $Sp(\pi) = \max_{\mathbb{P}_t \in \pi}(Sp(\mathbb{P}_t))$ of any configuration in the derivation. Taking the minimum space over all PCR refutations of a formula F, we obtain the monomial space complexity $Sp(F \vdash \bot) = \min_{\pi : F \vdash \bot}(Sp(\pi))$ of refuting F in PCR.

As already mentioned above, we can assume without loss of generality that all polynomials in a PCR derivation are multilinear, since higher powers of variables can be eliminated by using the Boolean axioms without increasing the monomial space by more than a constant factor.

In the rest of this paper, when we discuss just "space" we mean monomial space. Any unsatisfiable CNF formula with m clauses over n variables has a refutation in space $\mathrm{O}(\min(m,n))$. Such an upper bound holds for clause space in resolution [13], and PCR simulates resolution efficiently with respect to space. In this paper, we give an exposition of the result that this upper bound is asymptotically tight for random CNF formulas with $m = \Theta(n)$ clauses, i.e., formulas sampled according to the following distribution.

Definition 2.2. A *random k-CNF* formula F with n variables and density Δ is the outcome of drawing Δn clauses with repetition uniformly and independently at random from all (non-trivial) k-clauses with n variables. We write $F \sim \mathcal{F}_k^{n,\Delta}$ to denote that F is a formula sampled from this distribution.

A sequence of events $\mathcal{E}_n$ is said to happen *asymptotically almost surely* if $\lim_{n \to \infty} \Pr[\mathcal{E}_n] = 1$. It is well known (and not hard to show) that for every positive integer k there is a constant Δ_k such that for $\Delta \geq \Delta_k$ it holds asymptotically almost surely that $F \sim \mathcal{F}_k^{n,\Delta}$ is unsatisfiable.

[4] We mention for completeness that in [1] monomial space is defined as counting *without* repetitions, which seems like a slightly less natural measure. In practice, the distinction is mostly moot, since all monomial space lower bounds are in terms of the number of distinct monomials.

A way to understand the hardness of refuting a CNF formula F is to study the expansion of the *clause-variable incidence graph* G_F of F. This is the bipartite graph $G_F = (U \dot{\cup} V, E)$ where the left vertices in U are labelled by the clauses C in F, the right vertices in V are labelled by the variables x in F, and there is an edge (C,x) if the variable x occurs in the clause C.

Definition 2.3 (Bipartite expander)**.** In a bipartite graph $G = (U \dot{\cup} V, E)$, the neighbourhood $N(A) = \{v \mid u \in A, (u,v) \in E\}$ of a subset of vertices $A \subseteq U$ is the set of all vertices in V that have an adjacent vertex in A. We say that $G = (U \dot{\cup} V, E)$ is an (s,δ)-*bipartite expander* if for all $A \subseteq U$ with $|A| \leq s$ it holds that $|N(A)| \geq \delta |A|$.

We note for the record that clause-variable incidence graphs of randomly sampled k-CNF formulas are guaranteed to be expanders almost surely as the number of variables goes to infinity.

Theorem 2.4 ([11])**.** *For every fixed $k \geq 4$ and $\Delta > 0$, there exist constants $\gamma, \delta > 0$ such that the clause-variable incidence graph G_F of a random k-CNF formula $F \sim \mathcal{F}_k^{n,\Delta}$ is a $(\gamma n, 2+\delta)$-bipartite expander asymptotically almost surely.*

3 A Lower Bound on PCR Space of Refuting Random k-CNF Formulas

We now present our alternative exposition of the result by Bonacina and Galesi [8] that randomly sampled 4-CNF formulas require asymptotically maximal, linear, monomial space to be refuted in PCR. This is a consequence of the more general statement that the space complexity of refuting a CNF formula is related to the expansion properties of its clause-variable incidence graph.

Our exposition simplifies the presentation in [8] and highlights the role of so-called *online matchings*. Recall that as discussed in Section 1.2, given a small-space PCR derivation $\pi = (\mathbb{P}_0, \mathbb{P}_1, \ldots, \mathbb{P}_\tau)$ we want to construct a sequence of *shadow configurations* $(\mathbb{D}_0, \mathbb{D}_1, \ldots, \mathbb{D}_\tau)$ such that for every t it holds that $\mathbb{D}_t$ is satisfiable and $\mathbb{D}_t \vDash \mathbb{P}_t$. Each shadow configuration $\mathbb{D}_t$ is a 2-CNF formula $\mathbb{D}_t = Q_t \wedge M_t$, where the *axiom part* Q_t consists of variable-disjoint subclauses of axiom clauses from the input CNF formula that have been downloaded in the derivation π. Another way of expressing this is that we maintain a matching between subsets of downloaded axioms and pairs of variables. Since the set of axiom clauses changes over time, this matching needs to be maintained dynamically, and so we need a notion of adaptive matchings.

This notion of dynamically changing matchings is captured by the following game played by two players on a bipartite graph (which in our case is the clause-variable incidence graph of the input CNF formula). Player P has some tokens, and at each round she can place one of them on some left vertex (i.e., a clause). Every time she does that, Player D must match that token with a right vertex in its neighbourhood (i.e., a variable in that clause), and he must choose a vertex that is not already matched to another token. At the end of each round Player P can also remove one or more tokens, which frees up the right vertices matched to these tokens. To make the game more interesting, we also impose the condition that at any point of play there can be at most r tokens on any single left vertex and at most s tokens placed on the graph in total.

At some point Player D may be unable to match a token, because there is no free right vertex available. On a clause-variable incidence graph, for example, Player P may place on a set of clauses more tokens than the number of variables occurring in these clauses. We say that Player P wins the game if she can force a position where Player D has no legal move. We say that Player D wins the came if he has a strategy that will allow him to play the game indefinitely. Such a strategy is what we define to be an *online matching*.

Definition 3.1 (Online matching)**.** Let $G = (U \dot\cup V, E)$ be a bipartite graph. A set of edges $L \subseteq E$ is an *r-matching* if every vertex in U is incident to at most r edges in L and every vertex in V is incident to at most one edge in L. An r-matching L is a *perfect r-matching* from $A \subseteq U$ into $B \subseteq V$ if every vertex in A is incident to exactly r edges in L and no vertex in $V \setminus B$ is incident to L.

A collection of r-matchings $\mathcal{L}$ is an (s,r)-*online matching* if

1. $\emptyset \in \mathcal{L}$;
2. $\mathcal{L}$ is closed under taking subsets, in the sense that for any $L \in \mathcal{L}$ and $L' \subseteq L$ it holds that $L' \in \mathcal{L}$;
3. for each r-matching $L \in \mathcal{L}$ such that $|L| < s$, and for each $u \in U$ incident to less than r edges in L, there exists a $v \in V$ such that $L \cup \{(u,v)\} \in \mathcal{L}$.

This notion is also known as a *depth-one wide-sense nonblocking limited generalized concentrator* [14] and, for $r = 1$, as a *matching game* [4]. In our exposition, we use Definition 3.1 instead of the double matching property employed in the original proof of the PCR monomial space lower bound for random k-CNF formulas [8].

Now we can state the main theorem of this paper, which connects the presence of online matchings in the incidence graph G_F of a CNF formula F with the PCR monomial space complexity of refuting F.

Theorem 3.2 ([8, Theorem 3.5])**.** *If the clause-variable incidence graph of a CNF formula F has an $(s,2)$-online matching, then PCR requires monomial space greater than $s/8$ to refute F.*

As a technical side remark, it might be worth noticing that if F would happen to be satisfiable, then the theorem above definitely holds, since there is no refutation of F regardless of the space complexity.

On the face of it, the requirement in Theorem 3.2 of online matchings for large s might seem very strong. However, it turns out that bipartite graphs with good enough expansion have online matchings [14].

Theorem 3.3 ([14, Proposition 1])**.** *If the graph G is an $(2s, r+\delta)$-bipartite expander, then G has a $(\delta s, r)$-online matching.*

The clause-variable incidence graph of a random k-CNF formula has good expansion almost surely by Theorem 2.4, and therefore has linear-size online matchings by Theorem 3.3. The result by Bonacina and Galesi [8] that random formulas require large PCR monomial space is then an immediate corollary of Theorem 3.2 (where we also get the minor improvement that the lower bound does not depend on k).

Corollary 3.4 ([8, Theorem 5.5])**.** *For every fixed $k \geq 4$ and $\Delta > 0$, a random k-CNF formula sampled from $\mathcal{F}_k^{n,\Delta}$ requires $\Omega(n)$ monomial space to be refuted in PCR asymptotically almost surely.*

We will spend the rest of this section proving Theorem 3.2. As outlined in the introduction, the plan is to define for each configuration in a purported PCR refutation a simpler and more structured auxiliary *shadow configuration* that implies it. If all configurations in the PCR refutation are small, then all such auxiliary configurations will be small and satisfiable. That contradicts the fact that the last auxiliary configuration implies the last (unsatisfiable) configuration in the PCR refutation.

Definition 3.5 (Shadow configuration)**.** A *shadow configuration* for a CNF formula F is a 2-CNF formula $\mathbb{D} = Q \wedge M$ satisfying the following properties:

1. Clauses in the *axiom part* Q are over pairwise disjoint sets of variables, and the same applies to clauses in the *derived part* M.

2. Each variable in each clause in M is shared with exactly one clause in Q, and each clause in Q shares exactly one variable with exactly one clause in M, thus associating each $D \in M$ with two unique clauses $C_1, C_2 \in Q$ in a perfect 2-matching between the clause sets.

3. Each clause in Q is a subclause of an axiom clause in F, with which it is associated in a bijective fashion.

The axiom part Q keeps track of which axioms we downloaded: in our proof, every clause in Q will be a subclause of some previously downloaded axiom in F. The derived part M overapproximates information that has been derived from axioms that might no longer be in memory. Observe that by definition $|Q| = 2 \cdot |M|$. As an example, the formula $Q \wedge M$ with

$$Q = \{\overline{a} \vee x,\, b \vee y,\, c \vee z,\, d \vee \overline{w}\} \tag{3.1a}$$

and

$$M = \{a \vee b,\, \overline{c} \vee d\} \tag{3.1b}$$

is a shadow configuration for the formula

$$F = \{\overline{a} \vee \overline{b} \vee x \vee \overline{y},\, b \vee \overline{c} \vee \overline{x} \vee y,\, a \vee b \vee c \vee z,\, d \vee x \vee \overline{y} \vee \overline{w}\}\ . \tag{3.1c}$$

Given a PCR derivation $\pi = (\mathbb{P}_0, \mathbb{P}_1, \ldots, \mathbb{P}_\tau)$, we build the shadow configurations $\mathbb{D}_0, \mathbb{D}_1, \ldots, \mathbb{D}_\tau$ such that $\mathbb{D}_t \vDash \mathbb{P}_t$ by forward induction. For each $\mathbb{P}_t$ in the refutation we build a new $\mathbb{D}_t$ based on the structure of $\mathbb{D}_{t-1}$ and considering the derivation step applied to go from $\mathbb{P}_{t-1}$ to $\mathbb{P}_t$. Informally, for an axiom download we find a clause with two literals that is variable-disjoint with $\mathbb{D}$ and add it to Q. For an inference step we do nothing. After an erasure step the shadow configuration still implies the configuration in the proof, but it may now be too large. Therefore we compress it into a smaller shadow configuration by means of the following locality lemma.

Lemma 3.6 (Locality lemma)**.** *Let $\mathbb{D} = Q \wedge M$ be a shadow configuration for a CNF formula F that implies a configuration $\mathbb{P}$. Then there exists a shadow configuration $\mathbb{D}' = Q' \wedge M'$ that also implies $\mathbb{P}$ and is such that $|M'| \leq 2 \cdot Sp(\mathbb{P})$ and $Q' \subseteq Q$.*

We defer the proof of this crucial lemma to the end of the section and show how what we now have is sufficient to establish the main theorem.

Proof of Theorem 3.2. Let G_F denote the clause-variable incidence graph of F and fix an $(s,2)$-online matching $\mathcal{L}$ for G_F. Let $\pi = (\mathbb{P}_0, \ldots, \mathbb{P}_\tau)$ be a derivation from F in monomial space $Sp(\pi) \leq s/8$. We show by forward induction over π that every configuration $\mathbb{P}_i$ of π is satisfiable. This is sufficient to prove such derivation is not a refutation.

To do this, we build a sequence of shadow configurations $(\mathbb{D}_0, \ldots, \mathbb{D}_\tau)$ such that for each $\mathbb{D}_t = Q_t \wedge M_t$ we have the invariants that:

(a) $\mathbb{D}_t$ is satisfiable;

(b) $\mathbb{D}_t$ implies $\mathbb{P}_t$; and

(c) the inequality $|M_t| \leq 2 \cdot Sp(\mathbb{P}_t)$ holds.

For each shadow configuration $\mathbb{D}_t = Q_t \wedge M_t$ we maintain a bijection between Q_t and a subset of axioms $A_t \subseteq F$ such that every clause in Q_t is a subclause of an axiom in A_t, as well as a perfect 2-matching L_t in G_F from A_t into $Vars(Q_t)$, where for a subclause $x^a \vee y^b$ of an axiom $C = x^a \vee y^b \vee C'$ (represented as the monomial $m_C = x^a y^b m_{C'}$) we have $L(m) = \{x,y\}$. The key element needed to maintain the invariants of the sequence is that the PCR derivation has small space. Notice that invariant (a) always holds, since we can first satisfy all clauses in M_t independently (they are defined on pairwise disjoint variables), and then we have at least one unassigned variable in each clause of Q_t, not shared with any other clause of Q_t.

Base case For the base case $\mathbb{P}_0 = \emptyset$ it is enough to pick $\mathbb{D}_0 = \emptyset$, $A_0 = \emptyset$, and $L_0 = \emptyset$.

Axiom download We have that $\mathbb{P}_t = \mathbb{P}_{t-1} \cup \{p\}$, where p is the monomial that encodes an axiom of F. If $\mathbb{D}_{t-1} \vDash p$, it is enough to pick $\mathbb{D}_t = \mathbb{D}_{t-1}$ as in the inference case below. In particular, this takes care of the download of Boolean axioms and complementary axioms.

Otherwise, we need to enlarge the matching and the shadow configuration. Since $\mathbb{D}_{t-1} \nvDash p$, the axiom clause encoded by p is not a superset of any clause in Q_{t-1} and therefore no such clause is in A_{t-1} either. By invariant (c), we have that

$$|L_{t-1}| = 2 \cdot |A_{t-1}| = 2 \cdot |Q_{t-1}| = 4 \cdot |M_{t-1}| \leq 8 \cdot Sp(\mathbb{P}_{t-1}) \leq 8 \cdot Sp(\mathbb{P}_t) - 8 \leq s - 8 \ . \tag{3.2}$$

Since G_F has an $(s,2)$-online matching by assumption, we can extend L_{t-1} to a 2-matching $L' \supseteq L_{t-1}$ that matches the axiom p to two variables $\{x,y\}$ in p not occurring in Q_{t-1}, and therefore, by definition, not occurring in $\mathbb{D}_{t-1}$. Observe that this also shows that $\mathbb{D}_{t-1} \wedge p$ is satisfiable.

A brief look at the definition reveals that a shadow configuration requires an even sized axiom part. To ensure that this technical condition is met, we simulate a "dummy" axiom download. Let q be another axiom such that $\mathbb{D}_{t-1} \wedge p \nvDash q$. Such an axiom must exist, or otherwise F is satisfiable and the theorem holds vacuously. Extending the matching again, which we can do because $|L'| \leq s - 6$, there are two new variables $\{z,w\}$ in q not present in $\mathbb{D}_{t-1} \cup \{x,y\}$ and a 2-matching $L_t \supseteq L'$ from $A_{t-1} \cup \{p,q\}$ into $Vars(Q_{t-1}) \cup \{x,y,z,w\}$.

Let $p' = x^a \vee y^b \subseteq p$ and $q' = z^c \vee w^d \subset q$ be the subclauses of p and q supported over variables $\{x,y\}$ and $\{z,w\}$ respectively. Set $Q_t = Q_{t-1} \cup \{p',q'\}$ and $M_t = M_{t-1} \cup \{x^a \vee z^c\}$. This completes the construction of $\mathbb{D}_t$,[5] and all the required invariants can be verified to hold by construction.

Inference It is enough to pick $\mathbb{D}_t = \mathbb{D}_{t-1}$, because $\mathbb{P}_{t-1} \vDash \mathbb{P}_t$ and $Sp(\mathbb{P}_t) > Sp(\mathbb{P}_{t-1})$.

[5] The exact choice of variables and even polarities for $x^a \vee z^c$ do not matter. At this point, the two axioms p and q are implied by Q_t, and the clause $x^a \vee z^c \in M_t$ is redundant. Later in the derivation, however, the monomial space of some configuration $\mathbb{P}_{t'}$ may shrink together with the corresponding shadow configuration (via Lemma 3.6). Then either p' or q' (or both) might be removed from the shadow configuration, and the clause $x^a \vee z^c$ might be used to construct a new clause that is needed to imply $\mathbb{P}_{t'}$.

Erasure Observe that $\mathbb{P}_{t-1} \vDash \mathbb{P}_t$ holds, just as in the inference case, but the monomial space of $\mathbb{P}_t$ is smaller than that of $\mathbb{P}_{t-1}$. Therefore $\mathbb{D}_{t-1} = Q_{t-1} \wedge M_{t-1}$ could be too large to be used as the shadow configuration at step t. We need to apply Lemma 3.6 in order to obtain a shadow configuration $\mathbb{D}_t = Q_t \wedge M_t$ that implies $\mathbb{P}_t$ and such that $|M_t| \leq 2Sp(\mathbb{P}_t)$. Since $Q_t \subseteq Q_{t-1}$ we can also take $A_t \subseteq A_{t-1}$ and $L_t \subseteq L_{t-1}$.

It follows by induction that in any PCR derivation $\pi = (\mathbb{P}_0, \ldots, \mathbb{P}_\tau)$ from F in monomial space at most $s/8$ all configurations $\mathbb{P}_t$ are satisfiable, which establishes the theorem. □

It still remains to prove the locality lemma, for which we will need Hall's marriage theorem.

Theorem 3.7 (Hall's marriage theorem). *A bipartite graph $G = (U \dot{\cup} V, E)$ has a perfect matching from U to V if and only if for all $U' \subseteq U$ it holds that $|N(U')| \geq |U|$.*

To establish the locality lemma, we use a straightforward corollary of Hall's marriage theorem as stated next.

Lemma 3.8. *A bipartite graph $G = (U \dot{\cup} V, E)$ has a perfect 2-matching from U to V if and only if for all $U' \subseteq U$ it holds that $|N(U')| \geq 2|U'|$.*

Proof. If a 2-matching exists, then this matching shows for all $U' \subseteq U$ that $|N(U')| \geq 2|U'|$. Conversely, if the neighbourhood condition holds, we can create two duplicates u', u'' of each vertex $u \in U$ with $N(u') = N(u'') = N(u)$, apply Hall's theorem to obtain a matching in the duplicate graph, and finally merge all duplicates left vertices u' and u'' back into u to recover a 2-matching. □

Proof of Lemma 3.6. Suppose for the shadow configuration $\mathbb{D} = Q \wedge M$ that $\mathbb{D} \vDash \mathbb{P}$. We need to construct another shadow configuration $\mathbb{D}' = Q' \wedge M'$ such that $\mathbb{D}' \vDash \mathbb{P}$, $Q' \subseteq Q$, and $|M'| \leq 2 \cdot Sp(\mathbb{P})$.

We begin our construction of $\mathbb{D}'$ by choosing which parts of $\mathbb{D}$ to keep. Towards this goal, we build a bipartite graph $H = (U \dot{\cup} V, E)$ where the left vertex set U is identified with the set of distinct monomials in $\mathbb{P}$ and the right vertex set V is identified with clauses in the derived part M. Recall that every clause $C \in M$ unambiguously identifies two clauses in Q, which we denote $Q(C)$, that share exactly one variable each with C. We add an edge (m, C) to H if the monomial m mentions any of the four variables in the two clauses $Q(C)$.

We split the left vertex set U of H into two parts as follows. We fix U_1 to be a set of maximal size such that $|N(U_1)| \leq 2 \cdot |U_1|$ and let $U_2 = U \setminus U_1$. We set $M_1 = N(U_1)$ and write $Q_1 = Q(M_1) = \bigcup_{C \in M_1} Q(C)$ to denote the $2 \cdot |M_1|$ clauses in Q that have variables in common with M_1, and let $\mathbb{D}_1 = Q_1 \wedge M_1$. Since $|M_1|$ is sufficiently small relative to U_1, we will keep the part $\mathbb{D}_1$ of $\mathbb{D}$ as is.

As a warm-up, let us consider the special case when $U_1 = U$, so that $U_2 = \emptyset$. (This case is actually covered by the general argument below, but it might still be helpful to consider it separately.) In this case it holds not only that $|M_1|$ is small enough but also that $\mathbb{D}_1 \vDash \mathbb{P}$, and so this shadow configuration matches the conclusions in the locality lemma. To see this, let α_1 be any assignment that satisfies $\mathbb{D}_1$. Since $Vars(\mathbb{D} \setminus \mathbb{D}_1) \cap Vars(\mathbb{D}_1) = \emptyset$, the assignment α_1 can be extended to a satisfying assignment α for all of $\mathbb{D}$, and since $\mathbb{D} \vDash \mathbb{P}$ it follows that α satisfies $\mathbb{P}$. But by construction it holds that $Vars(\mathbb{D} \setminus \mathbb{D}_1) \cap Vars(\mathbb{P}) = Vars(\mathbb{D} \setminus \mathbb{D}_1) \cap Vars(\mathbb{D}_1) = \emptyset$ if $U_1 = U$, so the truth value of $\mathbb{P}$ is decided by α_1. Hence any satisfying assignment to $\mathbb{D}_1$ must also satisfy $\mathbb{P}$ if $U_1 = U$.

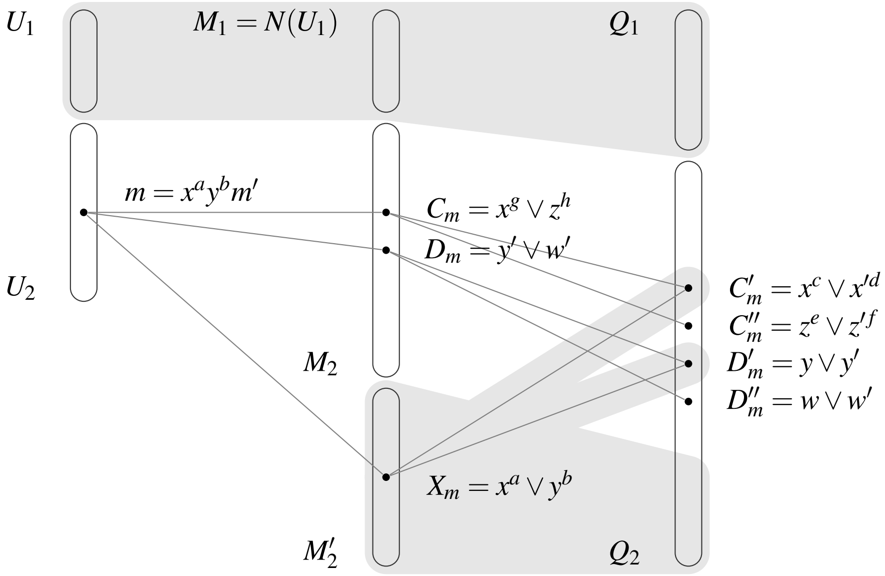

Figure 1: Building a smaller shadow configuration in the proof of Lemma 3.6.

For the general case when $U_2 = U \setminus U_1 \neq \emptyset$, we let $M_2 = M \setminus M_1$. Observe that any non-empty set $A \subseteq U_2$ must satisfy $|N(A) \cap M_2| > 2 \cdot |A|$. This is so since otherwise A could be added to U_1 to obtain a larger set $U'_1 = U_1 \cup A$ such that $|N(U'_1)| \leq 2 \cdot |U'_1|$, contradicting that U_1 is a maximal-size set with this property. From this it in turn follows by Lemma 3.8 that there is a perfect 2-matching from U_2 into M_2.[6] We fix such a 2-matching L to be used in our construction of the second part $\mathbb{D}_2 = Q'_2 \wedge M'_2$ of the new shadow configuration $\mathbb{D}'$. To understand the technical details of this construction, it may be helpful to use Figure 1 as a reference.

Consider any monomial $m \in U_2$. The 2-matching L associates to m two clauses $L(m) = \{C_m, D_m\}$ in M_2. Each of them corresponds in turn to two clauses in $Q \setminus Q_1$, which we denote $Q(C_m) = \{C'_m, C''_m\}$ and $Q(D_m) = \{D'_m, D''_m\}$. By construction, the monomial m has some variable x in common with either C'_m or C''_m (but not necessarily with C_m), and another variable y in common with either D'_m or D''_m (but not necessarily with D_m). Writing the monomial m as $m = x^a y^b m'$ for some $a, b \in \{0, 1\}$, we define a new clause $X_m = x^a \vee y^b$, with the same polarities of x and y as in the monomial m, to be added to M'_2. Note that this means that any assignment that satisfies X_m will make m evaluate to 0 and vanish. By construction, the clause X_m has exactly one variable in common with exactly one of the clauses C'_m and C''_m and another one in common with either D'_m or D''_m. We let $Q(X_m)$ denote these two clauses in $\{C'_m, C''_m, D'_m, D''_m\}$, which are to be added to Q'_2. (In Figure 1, we have $Q(X_m) = \{C'_m, D'_m\}$.)

Once we are done processing every monomial $m \in U_2$ in this way, we set $M'_2 = \bigcup_{m \in U_2} \{X_m\}$ and $Q'_2 = \bigcup_{m \in U_2} Q(X_m)$, and define the new shadow configuration to be $\mathbb{D}' = M' \wedge Q'$ for $M' = M_1 \wedge M'_2$ and $Q' = Q_1 \wedge Q'_2$. Essentially, we start with $\mathbb{D}$ and then replace M_2 and Q_2 with M'_2 and Q'_2, respectively, to obtain a shadow configuration $\mathbb{D}'$ that we claim satisfies the conclusion of the locality lemma.

[6] In particular, this means that any monomial of degree 0 (i.e., the empty monomial 1) or degree 1 must be part of U_1.

It is straightforward to check that $Q' \subseteq Q$ and that the size condition

$$|M'| = |M_1| + |M_2'| \leq 2 \cdot |U_1| + |U_2| \leq 2 \cdot Sp(\mathbb{P}) \tag{3.3}$$

is fulfilled. To show that $\mathbb{D}' \vDash \mathbb{P}$, we argue indirectly as follows:

- Let α be any truth value assignment that satisfies $\mathbb{D}'$.
- Using α, we construct another truth value assignment β such that
  - β satisfies $\mathbb{D}$ and
  - β assigns the same value as α to all monomials m in $\mathbb{P}$.
- Since $\mathbb{D} \vDash \mathbb{P}$, we have that β satisfies $\mathbb{P}$.
- But then α must also satisfy $\mathbb{P}$, since α and β agree on all monomials in $\mathbb{P}$, and since α is an arbitrary truth value assignment satisfying $\mathbb{D}'$ it follows that $\mathbb{D}' \vDash \mathbb{P}$ as claimed.

To implement this plan, fix any assignment α to $Vars(\mathbb{D}) \cup Vars(\mathbb{D}')$ satisfying $\mathbb{D}'$. Define the assignment β_1 on $Vars(\mathbb{D}')$ to agree with α. Note that α and β_1 agree on all monomials in U_1, since $Vars(U_1) \cap Vars(\mathbb{D}') \subseteq Vars(\mathbb{D}_1)$ by construction.

To construct a truth value assignment β_2 on $Vars(\mathbb{D}) \setminus Vars(\mathbb{D}')$, consider any clause $C \in M \setminus M' \subseteq M_2$ and let $Q(C) = \{C', C''\}$ be the two clauses in Q_2 associated with C. Recall that by the definition of shadow configurations we have $|Vars(C) \cap Vars(C')| = |Vars(C) \cap Vars(C'')| = 1$ and $Vars(C') \cap Vars(C'') = \emptyset$. Notice also that neither C' nor C'' has any variable in common with any monomial in U_1 by construction. We have two cases, depending on whether C was matched to some $m \in U_2$ by L or not:

1. If the edge (m, C) is in the 2-matching L, then we have that $|Vars(C) \cap Vars(X_m)| \leq 1$ and that exactly one of the clauses C' and C'' is in Q'. Let us say it is the former without loss of generality. Then we let β_2 assign one of the variables in $Vars(C) \setminus Vars(X_m)$ to satisfy C and then the free variable in $Vars(C'') \setminus Vars(C)$ to satisfy C''. Note that the clause C' is in $Q' \subseteq \mathbb{D}'$ and is hence satisfied by α. (Referring to Figure 1 again, the reader can take $C = C_m = x^g \vee z^h$, $C' = C_m' = x^c \vee x'^d$, and $C'' = C_m'' = z^e \vee z'^f$ to be example clauses for this argument. In this case we can choose $\beta_2(z) = h$ to satisfy C and $\beta_2(z') = f$ to satisfy C''.)

2. If the clause C is not in the 2-matching L, then all four variables in C' and C'' are unassigned by α, and we can choose for β_2 first to assign a variable so as to satisfy C and then pick values for the variables in $Vars(C') \setminus Vars(C)$ and $Vars(C'') \setminus Vars(C)$ to satisfy C' and C''. (Looking at Figure 1 and assuming—contrary to what is illustrated there—that $C = D_m$ is not in the matching L, we could implement this procedure by letting β_2 set the three variables y', y, and w to true to be guaranteed to satisfy all three clauses $D_m = y' \vee w'$, $D_m' = y \vee y'$, and $D_m'' = w \vee w'$.)

Finally, in order to obtain a total assignment, we decide (just for definiteness) that β_2 should agree with α on any variable in $Vars(\mathbb{D}) \setminus Vars(\mathbb{D}')$ for which a value has not yet been chosen, and then set $\beta = \beta_1 \cup \beta_2$, which we can do since the latter two assignments are over disjoint set of variables. (This means that β agrees with α on all variables in the shaded area of Figure 1.)

To conclude the proof, let us argue that $\beta = \beta_1 \cup \beta_2$ is a truth value assignment as described in our plan above. Observe that β_1 satisfies $\mathbb{D} \cap \mathbb{D}' = \mathbb{D}_1$ since it agrees with α there, and that β_2 satisfies $\mathbb{D} \setminus \mathbb{D}'$ by design. In other words, $\beta = \beta_1 \cup \beta_2$ is a satisfying assignment for $\mathbb{D}$ and hence also for $\mathbb{P}$. As already noted above, β_1 and α agree on all monomials in U_1 by construction. Both β_2 and α satisfy M_2', which means that $\beta_2(m) = \alpha(m) = 0$ for all monomials in U_2 as also noted above. Thus, β and α assign the same values to all monomials $U = U_1 \cup U_2$ in $\mathbb{P}$, and it follows that α must satisfy $\mathbb{P}$. This concludes our proof that $\mathbb{D}' \vDash \mathbb{P}$, and the locality lemma follows. □

4 Concluding Remarks

In this paper we present an alternative proof of the result by Bonacina and Galesi [8] that refuting random 4-CNF formulas requires linear monomial space in polynomial calculus asymptotically almost surely. Our exposition is simpler in that we avoid the powerful, but somewhat abstract, combinatorial machinery in [8] and instead give an explicit construction of a sequence of 2-CNF formulas that imply the corresponding configurations in a polynomial calculus derivation. These 2-CNF formulas have a nice, intuitive interpretation in that they consist partly of subclauses of axioms currently in memory and partly of clauses tracing derivations made from axioms previously in memory.

We want to emphasize that the general framework developed by Bonacina and Galesi [8] is more powerful than the techniques presented in this paper in that it can be used not only to prove lower bound for random CNF formulas but also to unify all previously shown lower bounds for polynomial calculus space. We hope and believe, however, that our way of presenting the lower bound for random CNF formulas, which is arguably the strongest result in [8], can make it accessible to a wider audience.

We also hope that a deeper understanding of the monomial space lower bound for refuting random CNF formulas can make it possible to attack other open problems concerning space in polynomial calculus. In contrast to many other lower bound techniques in proof complexity, the polynomial calculus space lower bounds are quite sensitive to transformations such as converting a formula to 3-CNF using auxiliary variables. Thus, while tight lower bounds on space are known for the standard encoding of *pigeonhole principle formulas*, the only known lower bounds on polynomial calculus space for the canonical 3-CNF version of these formulas are those that follow from the resolution width lower bounds together with the results by Galesi, Kołodziejczyk, and Thapen [18], which incurs a square root loss comparing to the results known for the original formulas. For somewhat related technical reasons there are also no tight lower bounds for *functional pigeonhole principle formulas*, not even in the standard encoding with wide axioms. Also, although for so-called *Tseitin formulas* (encoding inconsistent systems of linear equations modulo 2) optimal space lower bounds have been shown for certain subclasses of graphs by Filmus et al. [16], the problem of establishing that any Tseitin formula over a d-regular expander graph requires linear space remains wide open. For small d, the only space lower bound known for polynomial calculus is the square root of the number of variables, which again follows by appealing to [18]. Quite recently, Austrin and Risse [3] proved that for large enough d_0, a Tseitin formula over an expander graphs of degree d_0 and n vertices requires $\Omega(n/\log n)$ space, which is almost tight.

Determining the space complexity in polynomial calculus also remains open for so-called *ordering principle formulas* as studied in, e.g., [19, 21], where the only space lower bounds that can be obtained are worse by a square root than what is known for resolution. And for *pebbling formulas* no monomial space lower bounds are known. Pebbling formulas are of particular interest since they exhibit extremal behaviour with respect to the clause space and width measures in resolution, having refutations in constant width but potentially requiring almost worst-case linear space except for a logarithmic factor as shown by Ben-Sasson and Nordström [5]. It would be very interesting to know whether these formulas have the same properties for monomial space versus degree in polynomial calculus.

This leads us to a final, very intriguing question. For resolution it is known that the width complexity of k-CNF formulas is a lower bound on the space complexity as shown by Atserias and Dalmau [2]. It is natural to ask whether the analogous result holds also for polynomial calculus, i.e., whether degree is a lower bound on monomial space. It was shown by Filmus et al. [16] that a lower bound on polynomial calculus space in terms of resolution width (which is a weaker measure than polynomial calculus degree) holds for certain types of substituted formulas. More recently, it was established by Galesi, Kołodziejczyk, and Thapen [18] that polynomial calculus space is at least the square root of the resolution width, as already alluded to above, but the question of whether this square root loss can be eliminated remains wide open.

Acknowledgements

We are indebted to Ilario Bonacina and Nicola Galesi for numerous and very useful discussions. We would also like to thank Bruno Bauwens, Eli Ben-Sasson, and Yuval Filmus for helpful conversations. Finally, we are grateful to the anonymous referees for helping us improve the presentation.

A Proof of Theorem 3.3 That Bipartite Expanders Have Online Matchings

In order to make this manuscript self-contained, in this appendix we reproduce the proof of Theorem 3.3 that bipartite graphs with good enough expansion have somewhat large online matchings. Our presentation follows [14] except for minor changes in notation and terminology.

Theorem A.1 (Theorem 3.3, restated)**.** *If the graph $G = (U \dot{\cup} V, E)$ is a $(2s, r+\delta)$-bipartite expander, then G has an $(\delta s, r)$-online matching.*

In order to prove that graph expansion implies the existence of online matchings, we need a few definitions. Given an r-matching L, we say that a right vertex $v \in V$ is *idle* with respect to L if it is not incident to any edge in L, and we write $N_L^{idle}(u)$ to denote the set of idle vertices with respect to L that are neighbours of $u \in U$. We use this notion to introduce several concepts relative to a set of left vertices $X \subseteq U$ and an r-matching L as follows:

- We write $N_L^{idle}(X) = \bigcup_{u \in X} N_L^{idle}(u)$ to denote the set of idle vertices with respect to L that are neighbours of some vertex in X.
- The *assets* $Ass_L(X) = |N_L^{idle}(X)|$ of X is defined to be the number of idle neighbours of X.
- The *liabilities* $Lia_L(X) = r \cdot |X| - |\{(u,v) \in L : u \in X\}|$ of X indicates how many more right vertices could be required for X in the matching game if the current position is L.

- The *balance* $Bal_L(X) = Ass_L(X) - Lia_L(X)$ of X simply subtracts the liabilities of X with respect to L from the assets.
- Finally, we say that X is *solvent* with respect to L if $Bal_L(X) \geq 0$, *critical* if $Bal_L(X) = 0$, and *bankrupt* with respect to L if $Bal_L(X) < 0$.

With this in hand, we define

$$\mathcal{L} = \{L : L \text{ is an } r\text{-matching such that for all } X \subseteq U,\ |X| \leq s, \text{ it holds that } Bal_L(X) \geq 0\} \tag{A.1}$$

to be the set of r-matchings L on G having the property that every left vertex subset $X \subseteq U$ of size at most s is solvent with respect to L. In order to prove that $\mathcal{L}$ is an online matching, we need to establish some properties of the balance function.

Claim A.2. *If it holds that $|L| < \delta s$, X is critical, and $|X| \leq 2s$, then in fact $|X| < s$.*

Proof. We have

$$Ass_L(X) = |N_L^{idle}(X)| \geq |N(X)| - |L| > (r+\delta)|X| - \delta s \ , \tag{A.2}$$

where we bound $|N(X)| \geq (r+\delta)|X|$ by expansion and $|L| < \delta s$ by hypothesis. Since $Lia_L(X) \leq r|X|$ always holds, if X is critical we can calculate that

$$0 = Bal_L(X) = Ass_L(X) - Lia_L(X) > (r+\delta)|X| - \delta s - r|X| \geq \delta(|X| - s) \ , \tag{A.3}$$

from which it follows that $|X| < s$. □

For our next claim, we recall that a function f mapping sets to real numbers is said to be *modular* if $f(A) + f(B) = f(A \cup B) + f(A \cap B)$ and *submodular* if $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$.

Claim A.3. *The balance $Bal_L(X)$ of X with respect to L is a submodular function in X.*

Proof. On the one hand, $Ass_L(X)$ is submodular. To see this, we can use the standard facts

$$N(X \cup Y) = N(X) \cup N(Y) \tag{A.4a}$$
$$N(X \cap Y) \subseteq N(X) \cap N(Y) \tag{A.4b}$$
$$|A \cup B| + |A \cap B| = |A| + |B| \tag{A.4c}$$

to compute that

$$\begin{aligned} Ass_L(X \cup Y) + Ass_L(X \cap Y) &= |N_L^{idle}(X \cup Y)| + |N_L^{idle}(X \cap Y)| && \text{(A.5a)} \\ &\leq |N_L^{idle}(X) \cup N_L^{idle}(Y)| + |N_L^{idle}(X) \cap N_L^{idle}(Y)| && \text{(A.5b)} \\ &= |N_L^{idle}(X)| + |N_L^{idle}(Y)| && \text{(A.5c)} \\ &= Ass_L(X) + Ass_L(Y) \ . && \text{(A.5d)} \end{aligned}$$

On the other hand, $Lia_L(X)$ is modular. This is because we can view this function as as the linear extension of the element-wise measure $Lia_L(X) = \sum_{u \in X} Lia_L(\{u\})$. Since a submodular function minus a modular function is submodular, it follows that $Bal_L(X) = Ass_L(X) - Lia_L(X)$ is a submodular function as claimed. □

Claim A.4. *If for two subsets X and Y of U of sizes $|X|,|Y| \leq s$ it holds that X and Y are both critical with respect to an r-matching $L \in \mathcal{L}$, then $X \cup Y$ is also critical with respect to L.*

Proof. We have

$$Bal_L(X \cup Y) \leq Bal_L(X) + Bal_L(Y) - Bal_L(X \cap Y) \leq -Bal_L(X \cap Y) \leq 0 \tag{A.6}$$

where the first inequality holds by the submodularity in Claim A.3, the second inequality is true by hypothesis (and is in fact an equality), and the third inequality holds because $X \cap Y$, having size at most s, is solvent with respect to $L \in \mathcal{L}$ as prescribed by the definition of $\mathcal{L}$ in Equation (A.1). □

Recall that for a Boolean expression P, the Iverson bracket $[\![P]\!]$ is the function

$$[\![P]\!] = \begin{cases} 1 & \text{if } P \text{ is true,} \\ 0 & \text{if } P \text{ is false.} \end{cases} \tag{A.7}$$

A straightforward case analysis shows that if the implication $P \to Q$ is always true, then the equality $[\![Q]\!] - [\![P]\!] = [\![\neg P \wedge Q]\!]$ holds.

Claim A.5. *Suppose that L is an r-matching such that $(u,v) \in L$ and let $K = L \setminus \{(u,v)\}$. Then it holds that $Bal_K(X) = Bal_L(X) + [\![u \notin X \wedge v \in N(X)]\!]$.*

Proof. Since v is idle with respect to K, we have the equality $Ass_K(X) = Ass_L(X) + [\![v \in N(X)]\!]$ for the assets of X with respect to K and L. For the liabilities, it holds that $Lia_K(X) = Lia_L(X) + [\![u \in X]\!]$. Putting this together, we obtain for the balance $Bal_K(X) = Ass_K(X) - Lia_K(X)$ that

$$Bal_K(X) = Bal_L(X) + [\![v \in N(X)]\!] - [\![u \in X]\!] = Bal_L(X) + [\![u \notin X \wedge v \in N(X)]\!] \ , \tag{A.8}$$

where the last equality follows from the argument presented just below Equation (A.7) since $u \in X$ implies $v \in N(X)$. □

We are now ready to prove that $\mathcal{L}$ as defined in Equation (A.1) is indeed a $(\delta s, r)$-online matching as claimed in Theorem 3.3, i.e., that $\emptyset \in \mathcal{L}$, that $\mathcal{L}$ is closed under taking subsets, and that for each r-matching $L \in \mathcal{L}$ such that $|L| < \delta s$ and each $u \in U$ incident to less than r edges in L there exists a vertex $v \in N(u)$ such that $L \cup \{(u,v)\} \in \mathcal{L}$.

Let us first show that $\emptyset \in \mathcal{L}$. Fix any set $X \subseteq U$ of size $|X| \leq s$. By expansion and since all vertices are idle we have $Ass_\emptyset(X) \geq (r+\delta)|X|$, while $Lia_\emptyset(X) = r|X|$ since the matching is empty. Hence, it holds that $Bal_\emptyset(X) = Ass_\emptyset(X) - Lia_\emptyset(X) > 0$ and any subset X of size at most s is solvent with respect to $\emptyset$, so $\emptyset \in \mathcal{L}$.

The fact that $\mathcal{L}$ is closed under taking subsets follows immediately from Claim A.5, since for any r-matchings $L \in \mathcal{L}$ and $K = L \setminus \{(u,v)\}$ and any subset $X \subseteq U$ we have $Bal_K(X) \geq Bal_L(X) \geq 0$.

Finally, let us argue how to extend a matching $L \in \mathcal{L}$ for any $u \in U$ incident to less than r edges in L. Assume for the sake of contradiction that for every idle vertex $v \in N_L^{idle}(u)$ it holds that $L \cup \{(u,v)\} \notin \mathcal{L}$, and let $X_v \subseteq U$ of size $|X_v| \leq s$ be a bankrupt set with $Bal_{L \cup \{(u,v)\}}(X_v) < 0$ witnessing this. Since by Equation (A.1) it holds that X_v is solvent with respect to L, i.e., $Bal_L(X_v) \geq 0$, and since we have that

$Bal_L(X_v) - Bal_{L\cup\{(u,v)\}}(X_v) = [\![u \notin X_v \wedge v \in N(X_v)]\!]$ by Claim A.5, it follows that $u \notin X_v$ and $v \in N(X_v)$ and also that $Bal_L(X_v) = 0$ so that X_v is in fact critical with respect to L.

Fix any pair $v', v'' \in N_L^{idle}(u)$, $v' \neq v''$, and consider $X_{v'} \cup X_{v''}$. By Claim A.4 it holds that $X_{v'} \cup X_{v''}$ is critical with respect to L, and since $|X_{v'} \cup X_{v''}| \leq |X_{v'}| + |X_{v''}| \leq 2s$ it then follows from Claim A.2 that $|X_{v'} \cup X_{v''}| < s$. Thus, if we define $X = \bigcup_{v\in N_L^{idle}(u)} X_v$ and apply Claim A.4 and Claim A.2 inductively, we can conclude that X is a critical set such that $|X| < s$. Jumping ahead a bit, the existence of this set will allow us to derive contradiction by showing that $L \in \mathcal{L}$ violates the conditions in Equation (A.1).

To this end, let $X' = X \cup \{u\}$. On the one hand, since for every $v \in N_L^{idle}(u)$ we have shown that $u \notin X_v$, it follows that $u \notin X = \bigcup_{v\in N_L^{idle}(u)} X_v$. Since u is incident to less than r edges in L by assumption, this in turn implies that

$$Lia_L(X') = Lia_L(X \cup \{u\}) = Lia_L(X) + Lia_L(\{u\}) > Lia_L(X) \quad . \tag{A.9}$$

On the other hand, we have also shown for every $v \in N_L^{idle}(u)$ that $v \in N(X_v)$, which means that $N_L^{idle}(u) \subseteq N_L^{idle}(X)$ so that

$$Ass_L(X') = Ass_L(X \cup \{u\}) = Ass_L(X) \tag{A.10}$$

holds. In other words, we have found a set X' of size $|X'| = 1 + |X| \leq s$ such that

$$Bal_L(X') = Ass_L(X') - Lia_L(X') < Ass_L(X) - Lia_L(X) = Bal_L(X) = 0 \quad , \tag{A.11}$$

i.e., X' is bankrupt with respect to L, but this contradicts that $L \in \mathcal{L}$. Hence, there must exist some vertex $v \in N_L^{idle}(u)$ such that $L \cup \{(u,v)\} \in \mathcal{L}$. This concludes the proof of Theorem 3.3.

References

[1] MICHAEL ALEKHNOVICH, ELI BEN-SASSON, ALEXANDER A. RAZBOROV, AND AVI WIGDERSON: Space complexity in propositional calculus. *SIAM Journal on Computing*, 31(4):1184–1211, 2002. Preliminary version in *STOC '00*. 2, 3, 4, 5, 6

[2] ALBERT ATSERIAS AND VÍCTOR DALMAU: A combinatorial characterization of resolution width. *Journal of Computer and System Sciences*, 74(3):323–334, May 2008. Preliminary version in *CCC '03*. 3, 15

[3] PER AUSTRIN AND KILIAN RISSE: Perfect matching in random graphs is as hard as Tseitin. *TheoretiCS*, 1:22:1–22:47, December 2022. Preliminary version in *SODA '22*. 14

[4] ELI BEN-SASSON AND NICOLA GALESI: Space complexity of random formulae in resolution. *Random Structures and Algorithms*, 23(1):92–109, August 2003. Preliminary version in *CCC '01*. 3, 8

[5] ELI BEN-SASSON AND JAKOB NORDSTRÖM: Short proofs may be spacious: An optimal separation of space and length in resolution. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS '08)*, pp. 709–718, October 2008. 3, 15

[6] PATRICK BENNETT, ILARIO BONACINA, NICOLA GALESI, TONY HUYNH, MIKE MOLLOY, AND PAUL WOLLAN: Space proof complexity for random 3-CNFs. *Information and Computation*, 255(1):165–176, August 2017. 2, 4

[7] ILARIO BONACINA: Total space in resolution is at least width squared. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP '16)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 56:1–56:13, July 2016. 2

[8] ILARIO BONACINA AND NICOLA GALESI: A framework for space complexity in algebraic proof systems. *Journal of the ACM*, 62(3):23:1–23:20, June 2015. Preliminary version in *ITCS '13*. 4, 5, 7, 8, 14

[9] ILARIO BONACINA, NICOLA GALESI, AND NEIL THAPEN: Total space in resolution. *SIAM Journal on Computing*, 45(5):1894–1909, January 2016. Preliminary version in *FOCS '14*. 2

[10] 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, pp. 233–350. IOS Press, 2nd edition, February 2021. Available at http://www.jakobnordstrom.se/publications/. 2

[11] VAŠEK CHVÁTAL AND ENDRE SZEMERÉDI: Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, October 1988. 7

[12] 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)*, pp. 174–183, May 1996. 2, 5

[13] JUAN LUIS ESTEBAN AND JACOBO TORÁN: Space bounds for resolution. *Information and Computation*, 171(1):84–97, 2001. Preliminary versions of these results appeared in *STACS '99* and *CSL '99*. 2, 3, 6

[14] PAUL FELDMAN, JOEL FRIEDMAN, AND NICHOLAS PIPPENGER: Wide-sense nonblocking networks. *SIAM Journal of Discrete Mathematics*, 1(2):158–173, May 1988. 8, 15

[15] YUVAL FILMUS, MASSIMO LAURIA, MLADEN MIKŠA, JAKOB NORDSTRÖM, AND MARC VINYALS: From small space to small width in resolution. *ACM Transactions on Computational Logic*, 16(4):28:1–28:15, July 2015. Preliminary version in *STACS '14*. 3

[16] YUVAL FILMUS, MASSIMO LAURIA, MLADEN MIKŠA, JAKOB NORDSTRÖM, AND MARC VINYALS: Towards an understanding of polynomial calculus: New separations and lower bounds. *Theory of Computing*, 21(4):1–48, August 2025. Preliminary version in *ICALP '13*. 14, 15

[17] YUVAL FILMUS, MASSIMO LAURIA, JAKOB NORDSTRÖM, NOGA RON-ZEWI, AND NEIL THAPEN: Space complexity in polynomial calculus. *SIAM Journal on Computing*, 44(4):1119–1153, August 2015. Preliminary version in *CCC '12*. 3, 4

[18] NICOLA GALESI, LESZEK KOŁODZIEJCZYK, AND NEIL THAPEN: Polynomial calculus space and resolution width. *Theory of Computing*, 21:6:1–67:29, September 2025. Preliminary version in *FOCS '19*. 14, 15

[19] NICOLA GALESI AND MASSIMO LAURIA: Optimality of size-degree trade-offs for polynomial calculus. *ACM Transactions on Computational Logic*, 12(1):4:1–4:22, November 2010. 14

[20] JAN KRAJÍČEK: *Proof Complexity*. Volume 170 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, March 2019. 2

[21] MLADEN MIKŠA AND JAKOB NORDSTRÖM: A generalized method for proving polynomial calculus degree lower bounds. *Journal of the ACM*, 71(6):37:1–37:43, November 2024. Preliminary version in *CCC '15*. 14

[22] JAKOB NORDSTRÖM: Narrow proofs may be spacious: Separating space and width in resolution. *SIAM Journal on Computing*, 39(1):59–121, May 2009. Preliminary version in *STOC '06*. 3

[23] JAKOB NORDSTRÖM: Pebble games, proof complexity and time-space trade-offs. *Logical Methods in Computer Science*, 9(3):15:1–15:63, September 2013. 3

[24] JAKOB NORDSTRÖM AND JOHAN HÅSTAD: Towards an optimal separation of space and length in resolution. *Theory of Computing*, 9:471–557, May 2013. Preliminary version in *STOC '08*. 3

[25] MICHAEL S. PATERSON AND CARL E. HEWITT: Comparative schematology. In *Record of the Project MAC Conference on Concurrent Systems and Parallel Computation*, pp. 119–127, 1970. 3

AUTHORS

Massimo Lauria
Associate Professor
Department of Statistical Science
Sapienza — Università di Roma
Rome, Italy
massimo.lauria@uniroma1.it
http://www.massimolauria.net/

Mladen Mikša
mladen.miksa@gmail.com

Jakob Nordström
Professor
Department of Computer Science
University of Copenhagen
Copenhagen, Denmark and
Department of Computer Science
Lund University
Lund, Sweden
jn@di.ku.dk
https://www.jakobnordstrom.se/

Marc Vinyals
Lecturer
School of Computer Science
Waipapa Taumata Rau — University of Auckland
Auckland, New Zealand
marc.vinyals@auckland.ac.nz
https://marcvinyals.gitlab.io/

ABOUT THE AUTHORS

MASSIMO LAURIA got his Ph. D. at the Department of Computer Science of University “La Sapienza” in Rome in 2009, advised by Nicola Galesi. After that he bounced around Europe (and even further) for postdocs and visiting positions between 2011 and 2017, in particular Prague, Stockholm, Tokyo and Barcelona, where he honed his research skills in complexity theory and proof complexity. After that he went back to Rome in 2017 to join the Department of Statistical Science at University “La Sapienza” in Rome, where he has been an associate professor since 2020.

MLADEN MIKŠA got his M. S. from the University of Zagreb in 2012, after which he did his Ph. D. at KTH Royal Institute of Technology, advised by Jakob Nordström, finishing in 2017. What happened after that is a mystery.

JAKOB NORDSTRÖM received his Ph. D. from KTH Royal Institute of Technology, advised by Johan Håstad. He has been a postdoc at MIT and an assistant and associate professor at KTH, and is now a full professor at the University of Copenhagen with a side affiliation across the Öresund bridge at Lund University. These days, he spends roughly half of his time proving exponential-time lower bounds for *NP*-hard combinatorial problems, and the other half designing applied algorithms that solve such problems in linear time.

MARC VINYALS did his undergraduate studies at UPC in Barcelona and his Ph. D. studies at KTH in Stockholm, advised by Jakob Nordström, and finishing in 2017. After visiting TIFR in Mumbai and the Technion in Haifa, he worked as a Docent at St Petersburg State University. He has been a Lecturer at Waipapa Taumata Rau — University of Auckland since 2023. His main research interests are proof complexity, communication complexity, and the theory of satisfiability solving. When he is not in front of a whiteboard, he can be found near a stage or in a kayak.