Electronic Colloquium on Computational Complexity, Report No. 57 (2025)

Multiparty Communication Complexity of Collision-Finding and Cutting Planes Proofs of Concise Pigeonhole Principles*

Paul Beame†
Allen School of Computer Science & Engineering
University of Washington

Michael Whitmeyer†
Allen School of Computer Science & Engineering
University of Washington

April 28, 2025

Abstract

We prove several results concerning the communication complexity of a collision-finding problem, each of which has applications to the complexity of cutting-plane proofs, which make inferences based on integer linear inequalities.

In particular, we prove an $\Omega(n^{1-1/k}\log k\ /2^k)$ lower bound on the k-party number-in-hand communication complexity of collision-finding. This implies a $2^{n^{1-o(1)}}$ lower bound on the size of tree-like cutting-planes refutations of the bit pigeonhole principle CNFs, which are compact and natural propositional encodings of the negation of the pigeonhole principle, improving on the best previous lower bound of $2^{\Omega(\sqrt{n})}$. Using the method of density-restoring partitions, we also extend that previous lower bound to the full range of pigeonhole parameters.

Finally, using a refinement of a bottleneck-counting framework of Haken and Cook and Sokolov for DAG-like communication protocols, we give a $2^{\Omega(n^{1/4})}$ lower bound on the size of fully general (not necessarily tree-like) cutting planes refutations of the same bit pigeonhole principle formulas, improving on the best previous lower bound of $2^{\Omega(n^{1/8})}$.

1 Introduction

The pigeonhole principle, which asserts that there is no injective function $f : [m] \to [n]$ for $m > n$, is a cornerstone problem in the study of proof complexity. It is typically encoded as unsatisfiable conjunctive normal form formula (CNF), henceforth denoted PHP_n^m, on the variables $y_{i,j}$, each of which is an indicator that "pigeon" i is mapped to "hole" j.

It is well known that any refutation of PHP_n^{n+1} using resolution proofs requires size $2^{\Omega(n)}$ [13] and the same asymptotic bound holds for all m that are $O(n)$ [6]. On the other hand, if we allow our proof system to reason about linear inequalities (for example using cutting-planes proofs), then it is easy to see that refuting PHP_n^{n+1} becomes easy – indeed, there exist polynomial size refutations of PHP_n^{n+1}.

Despite the pigeonhole principle having short cutting-planes refutations, the related clique-coloring formulas, which state that a graph cannot have both k-cliques and $k-1$-colorings, requires exponential-size cutting-planes refutation [26].[1] The clique-coloring formula can be viewed as a kind of indirect pigeonhole principle: The k nodes of the clique correspond to the pigeons and $k-1$ colors correspond to the holes, but the representation of possible mappings is quite indirect.

It is natural to wonder about the extent to which indirection is required for the pigeonhole principle to be hard for cutting-planes reasoning. As part of studying techniques for cutting-planes proofs, Hrubeš

*A preliminary version containing one of these results appears in [3]. An extended abstract of the current paper will appear in the proceedings of ICALP 2025.

†Research supported by NSF grants CCF-2006359 and CCF-2422205

[1] Lower bounds for restricted cutting-planes refutations of these formulas were earlier shown in [18, 4]

ISSN 1433-8092

and Pudlák [17] considered a very natural compact and direct way of expressing the pigeonhole principle, known as the *bit* or *binary* pigeonhole principle[2]. The bit pigeonhole principle analog of PHP_n^m (henceforth denoted BPHP_n^m) has $m \log n$ variables $x_{i,j}$ for $i \in [m], j \in [\log n]$ and the principle asserts that, when we organize these variables as an $m \times [\log n]$ matrix, the rows of the matrix all have distinct values. BPHP_n^m is the following CNF formula: for each $i \neq j \in [m]$, include the clauses of a CNF encoding that $x_i \neq x_j$. One can achieve this by including a clause for each $\alpha \in \{0,1\}^{\log n}$ expressing that $x_i \neq \alpha \vee x_j \neq \alpha$. The end result is a CNF with $\binom{m}{2} n$ clauses of size $2 \log n$.

Using techniques related to those of [26], Hrubeš and Pudlák [17] showed that BPHP_n^m requires cutting-planes refutations of size $2^{\Omega(n^{1/8})}$ for any $m > n$, proving that even a very direct representation of the pigeonhole principle is hard for cutting-planes proofs. Their arguments, like those of Pudlák, also apply to any proof system that has proof lines consisting of integer linear inequalities with two antecedents per inference that are sound with respect to 01-valued variables; such proofs are known alternatively as *semantic* cutting-planes proofs or $\mathsf{Th}(1)$ proofs [2].

Recently, Dantchev, Galesi, Ghani, and Martin [8] exhibited a $2^{\Omega(n/\log n)}$ lower bound on the size of any general resolution refutation of BPHP_n^m for all $m > n$. In fact, they showed that BPHP_n^m requires proofs of size $2^{\Omega(n^{1-\varepsilon})}$ for a more powerful class of proof systems that extend resolution by operating on k-DNFs (known as $\mathrm{Res}(k)$ proofs) for $k \leq \log^{1/2-\varepsilon'} n$. (Note that any sound proof system operating on DNFs requires size at least $2^{n^{\Omega(1)}}$ to refute PHP_n^{n+1} [25, 21, 15].) In addition, [8] showed that BPHP_n^m has no refutations in the Sherali-Adams proof system [29] of size smaller than $2^{\Omega(n/\log^2 n)}$. Finally, just as PHP_n^m has polynomial-size Sum-of-Squares refutations [12], Dantchev et al. showed that BPHP_n^m has polynomial-sized Sum-of-Squares refutations.

Given the large lower bounds for resolution, $\mathrm{Res}(k)$, and Sherali-Adams refutations of BPHP_n^m, it is natural to ask the extent to which the sub-exponential lower bounds can be improved for cutting-planes proofs; how close to a $2^{\Omega(n)}$ lower bound is possible?

1.1 Tree-like Proofs and Multiparty Communication

In prior work there has been progress towards this question for the restricted class of *tree-like* refutations. Tree-like proofs require that any time an inequality is used, it must be re-derived (i.e., the underlying graph of deductions is a tree); the polynomial-size cutting-planes refutations of PHP_n^{n+1} can be made tree-like. In contrast, Itsykson and Riazanov [19] showed that BPHP_n^m requires tree-like cutting-planes refutations of size $2^{\Omega(\sqrt{n})}$ when $m \leq n + \sqrt{n}$.

Our first result pushes this bound almost to its limit. Specifically, we prove that any tree-like semantic cutting-planes refutation of BPHP_n^m requires size $2^{n^{1-o(1)}}$ whenever $m \leq n + 2^{2\sqrt{\log n}-2}$.

In order to show this, we utilize a well-known connection between tree-like refutations and communication complexity. While the results of [19] for cutting planes rely on two-party communication complexity (and number-on-forehead multiparty communication for other results that we mention below), our stronger results are based on multiparty number-in-hand communication. In particular they are based on a similar natural collision-finding communication problem $\mathsf{Coll}_{m,\ell}^k$, in which each player $p \in [k]$ in the number-in-hand model receives an input in $x^{(p)} \in [\ell]^m$, and their goal is to communicate and find a pair $i \neq j \in [m]$ such that $x_i^{(p)} = x_j^{(p)}$ for all players $p \in [k]$. Such a communication problem is well-defined (in the sense that such a pair i, j always exists) when $m > \ell^k$.

This collision-finding problem is intimately related to the unsatisfiable BPHP_n^m formula via the following natural search problem associated with any unsatisfiable CNF formula: Given unsatisfiable CNF φ, the associated search problem Search_φ takes as input a truth assignment α to the variables of φ and requires the output of the index of a clause of φ that is falsified by α. In particular the connection follows by considering a natural k-party number-in-hand communication game that we denote by $\mathsf{Search}_\varphi^k$ wherein the assignment α to the variables of φ is evenly distributed among the k players. and the players must communicate to find an

[2]This encoding of the pigeonhole principle was introduced in [1].

answer for $\mathsf{Search}_{\varphi}(\alpha)$. It is not hard to see that if we have a communication protocol solving $\mathsf{Search}^k_{\mathrm{BPHP}^m_n}(\alpha)$ then such a protocol also solves $\mathsf{Coll}^k_{m,n^{1/k}}$ on input α.

Our first result is a lower bound on $\mathsf{Coll}^k_{m,n^{1/k}}$ that holds even when we allow randomized protocols.

Theorem 1.1. *The randomized number-in-hand communication complexity of* $\mathsf{Coll}^k_{m,n^{1/k}}$ *is* $\Omega(n^{1-1/k}\log k\ /2^k)$ *whenever* $n+1 \le m \le n+2^{k-2}n^{1/k}$.

We pause here to note that this bound is nearly tight. There is a deterministic protocol wherein the first player sends a subset of coordinates of size $\lceil m/n^{1/k}\rceil$ in which their inputs are all equal. This requires $\log\binom{m}{m/n^{1/k}} \lesssim (m/n^{1/k})\log(m/n^{1/k})$ bits; when $m \approx n$, this is $O(n^{1-1/k}\log(n^{1/k})) = O(n^{1-1/k}\log n\ /k)$ bits of communication. Player two then announces a subset of these coordinates on which they are equal of size $\lceil m/n^{2/k}\rceil$. The players can continue in this manner until they have found a collision (which is guaranteed by the pigeonhole principle). Note that the amount of communication is bounded by a geometric series, and is dominated by the first term, which results in communication $O(n^{1-1/k}\log n\ /k)$. This shows that up to logarithmic factors and a factor of 2^k, Theorem 1.1 is tight.

We state here a simplified corollary[3] of Theorem 1.1 which formalizes our lower bounds for cutting-planes refutations of BPHP^m_n.

Theorem 1.2. *Any tree-like semantic cutting-planes refutation of* BPHP^m_n *requires size* $2^{n^{1-2/\sqrt{\log n}-o(1/\sqrt{\log n})}}$ *when* $m \le n+2^{2\sqrt{\log n}-2}$.

We remark that Itsykson and Riazanov [19] utilized the same connection between communication and proof complexity to achieve their results. They were also interested in a k-party number-on-forehead version of $\mathsf{Coll}^k_{m,\ell}$ (in particular, in their version, the matrices are added rather than concatenated), which leads to weaker lower bounds in stronger proof systems $\mathsf{Th}(k-1)$ that manipulate degree $k-1$ polynomial inequalities.

Itsykson and Riazanov also left as an open problem whether their bounds for $\mathsf{Search}_{\mathrm{BPHP}^m_n}$ could be extended to the regime of the "weak" pigeonhole principle when $m = n+\Omega(n)$. Göös and Jain [10] first answered this in the affirmative, giving an $\Omega(n^{1/12})$ lower bound on the randomized communication complexity of $\mathsf{Coll}^2_{2n,n^{1/2}}$. Yang and Zhang [34] subsequently improved this to an $\Omega(n^{1/4})$ bound, which is tight for randomized computation. Because of the connection between communication protocols and tree-like communication, Yang and Zhang's bound showed that any tree-like cutting planes refutation of BPHP^m_n requires size $2^{\Omega(n^{1/4})}$. We extend the lower bounds of Yang and Zhang for all $k \ge 2$ as follows:

Theorem 1.3. *For all* $m > n$ *and* $2 \le k \le \log n\ /4$, *the randomized number-in-hand communication complexity of* $\mathsf{Coll}^k_{m,n^{1/k}}$ *is* $\Omega(n^{1/2-1/(2k)}/k)$.

This bound, whose proof combines techniques of Yang, Zhang, and Wang [33, 34], is much weaker than that of Theorem 1.1 when m is close to n but it is tight, up to an $O(\log n)$ factor, for randomized protocols when m is $n+\Omega(n)$. For $k = \Omega(\log n)$, it also implies the following proof complexity bound which asymptotically matches the bound of [19], but applies for all $m > n$.

Corollary 1.4. *For all* $m > n$, *any tree-like semantic cutting-planes refutation of* BPHP^m_n *requires size* $2^{\Omega(\sqrt{n})}$.

1.2 General Proofs and DAG-like Protocols

Our second result improves on the $2^{\Omega(n^{1/8})}$ lower bound on the size of cutting-planes refutations of BPHP^m_n (for all $m > n$) of Hrubeš and Pudlák [17], and subsumes the $2^{\Omega(n^{1/4})}$ lower bound for tree-like proofs of Yang and Zhang [34]. Hrubeš and Pudlák used the method of interpolation involving a version of the method of approximation due to Jukna [20]. On the other hand, we use a bottleneck-counting method inspired by recent work of Sokolov [31] that refines a method introduced by Haken and Cook [14].

[3] When m is somewhat larger, we can obtain somewhat weaker lower bounds.

Theorem 1.5. *Any semantic cutting planes refutation of* BPHP_n^m *requires size at least* $2^{n^{1/4}/\sqrt{2}-2}$*, for any* $m > n$*.*

We prove Theorem 1.5 via a connection between cutting-planes refutations and certain types of DAG-like communication protocols [16, 30]. Specifically, we prove a lower bound on the size of any two party *triangle-DAG* protocol computing $\mathsf{Search}_{\text{BPHP}_n^m}$. We define such protocols formally in Section 2.3.

For now, we state the connection as the following proposition, originally due to Sokolov and Hrubeš and Pudlák [30, 16]. Its proof can be found in Section 2.3 for completeness.

Proposition 1.6. *[30, 16] Given a semantic cutting planes refutation for CNF formula* φ *of size* s *and any partition of the variables of* φ *into two sets, there is a size* s *triangle-DAG computing* Search_φ*.*

In our setting, for BPHP_n^m, there is a natural partition of the variables such that each clause has exactly half of its variables coming from V_x and half of its variables coming from V_y. This can be achieved, for example, by letting V_x be the set of variables corresponding to the first $\frac{1}{2}\log n$ columns of the matrix associated with BPHP_n^m. In light of Proposition 1.6, we derive Theorem 1.5 by proving the following.

Theorem 1.7. *For the natural partition of variables* $V_x \sqcup V_y = [m \log n]$*, any triangle-DAG computing* $\mathsf{Search}_{\text{BPHP}_n^m}$ *requires size* $2^{n^{1/4}/\sqrt{2}-2}$*.*

2 Preliminaries

2.1 Proof Complexity

Proof complexity studies how the size required for refutations of unsatisfiable formulas depends on the size of the formulas themselves. The size of a refutation in general depends on the allowable structure (lines, derivation rules, etc.) of a proof. In general, a proof *system* corresponds to a polynomial-time verifier that can check proofs of a certain format. It generally suffices to derive refutations of unsatisfiable CNF formulas; so it is usual to focus solely on inputs of this form.

For most proof systems, a sequence of deductions can be thought of as a directed graph, where two (or possibly more) lines (whether given or derived) are combined soundly to create a new line. The underlying graph then has edges pointing from the derived inequality to its antecedents.[4] We say that a proof is *tree-like* if every inequality is used as an antecedent at most once in the proof – that is, if we want to use an inequality twice, we must derive it twice.

For example, in the *resolution* proof system the lines are clauses, and we have the derivation rule $(A \vee x) \wedge (B \vee \neg x) \vdash A \vee B$.

A more powerful proof system is the cutting planes proof system[5], denoted CP, where lines are linear inequalities. Clauses can be trivially converted into linear inequalities[6] but, more generally, cutting planes can start with any system of inequalities $Ax \leq b$ where A is a matrix with integer entries. We say that the system is unsatisfiable iff it is unsatisfiable for any $x \in \{0,1\}^n$. The most basic form of the cutting planes proof system consists of three rules: addition of inequalities, multiplication of inequalities by positive integers, and most crucially, the rounded division rule, also known as the Gomory-Chvátal rule. The rounded division rule is the simple but powerful observation that if $a_1, \ldots, a_n$ are all integers with a common factor c, and we have the inequality $a^T x \leq b$, then we can derive that $\frac{1}{c} a^T x \leq \lfloor \frac{b}{c} \rfloor$.

In general, there are many more sound derivation rules for integer/linear inequalities than just rounded division (such as saturation [9] for example), and even more generally, one may allow *any* sound derivation rule for linear inequalities, which yields what is known as *semantic* cutting planes or $\mathsf{Th}(1)$ – we use the two names interchangeably.

There is a well-known connection between communication complexity and tree-like proofs, which we will now detail. Given any unsatisfiable formula φ, an assignment α to the variables can be distributed

[4] The direction of arrows in this digraph may seem counterintuitive, but it is convenient when thinking of the graph as a search problem for a violated axiom. In this case, we can follow a path in the graph to find such a a violated axiom on one of the leaves/sources.

[5] Cutting planes proofs can simulate resolution proofs, see e.g. [20, 27].

[6] For example, $x \vee y \vee z$ could be converted to the inequality $x + y + z \geq 1$.

among k players, who must then communicate in order to find a a violated clause in φ. This is a search problem, and is denoted $\mathsf{Search}_\varphi(\alpha)$. Short tree-like proofs of the unsatisfiability of φ can often be converted into short protocols for Search_φ using standard techniques.

For example, a short tree-like proof of unsatisfiability of φ using the resolution rule naturally corresponds to a decision tree for finding a violated clause of φ in the following way. For every derivation of the form $(A \vee x) \wedge (B \vee \neg x) \vdash (A \vee B)$ where $A \vee B$ is known to be false, we query x in order see whether $A \vee x$ or $B \vee \neg x$ is necessarily false. We can continue in this manner, from the root of the tree-like refutation, until we hit an unsatisfied clause in the original formula.

On the other hand, tree-like semantic cutting planes refutations naturally correspond to *threshold* decision trees, which we now define.

Definition 2.1 (Threshold Decision Tree)**.** *A* threshold decision tree *is a tree whose vertices are labeled with inequalities of the form $a_1x_1 + \cdots a_nx_n \leq b$ where $a_1, \ldots, a_n, b$ are integers. Edges are labelled with 0 or 1, and leaves are axioms of a system of inequalities $Ax \leq b$.*

We traverse a threshold decision tree by computing the threshold function at the root, following the corresponding edge, and continuing in this manner until we hit a leaf. We say that a threshold decision tree computes the search problem for a formula φ if this process leads to a leaf corresponding to a violated clause in φ.

First, we have the following well-known lemma, which states that one can derive a low-depth threshold decision tree from a small $\mathsf{Th}(1)$ refutation.[7]

Proposition 2.2. *Given a size S tree-like $\mathsf{Th}(1)$ refutation of an unsatisfiable system $Ax \leq b$, there is a depth $O(\log S)$ threshold decision tree finding a violated axiom.*

Proposition 2.2 goes back to the work of Impagliazzo, Pitassi, and Urquhart [18], and can also be found for instance in [20]. We omit the proof, but the idea is a common one: find a node in the tree with roughly half (between $1/3$ and $2/3$) of the leaves as its descendants, make that the root of the threshold decision tree, and recurse.

2.2 Communication Complexity

We mainly focus on k-party number-in-hand communication, wherein each player p receives an input $x^{(p)}$, and the players' goal is to communicate as little as possible in order to compute a known function or relation involving their collective inputs. In general, players may have access to shared randomness, and we allow incorrect answers with probability $1/3$.

An important function for us is the number-in-hand disjointness problem with k players and input size n, henceforth denoted DISJ_n^k. This is the communication problem wherein each player and input in $\{0,1\}^n$, and they must decide if there exists a coordinate i for which they all have a 1 in that coordinate. Disjointness in general is an extremely well-studied problem [7, 27], and for the specific case of the NIH model, we have the following lower bound due to Braverman and Oshman.

Theorem 2.3 ([5])**.** *The randomized communication complexity of DISJ_n^k is $\Omega(n \log k)$.*

Our results rely on the following connection between threshold decision trees for finding violated clauses and k-party NIH communication. It is closely related to previous work.

Lemma 2.4. *For $x \in \{0,1\}^n$, if an unsatisfiable system $Ax \leq b$ on has a threshold decision tree of depth $d \leq n$ finding a violated axiom, then for any partition of the input variables into k parts there is a randomized protocol for $\mathsf{Search}_{NIH}^k(Ax \leq b)$ using $O(dk \log k \log n)$ bits of communication.*

[7] As noted in [20], there is no meaningful converse to this statement, since if there are m inequalities in our unsatisfiable system, there exists a trivial depth m threshold decision tree finding one that is violated.

Lemma 2.4 is similar for example to Lemma 5 in [18] or Lemma 19.11 in [20], but is slightly stronger, so we include a proof. The idea is that the players can use the given threshold decision tree to construct a protocol. Without loss of generality, using a well-known theorem of Muroga [23] without communication the players can replace each of the inequalities in the decision tree with an equivalent one over the Boolean hypercube with coefficients that are not too large.

The players then start at the root and evaluate each associated inequality using an efficient randomized protocol to evaluate each threshold function with high probability and moving to the appropriate child node until it reaches a leaf. Such a protocol dates back to Nisan [24] and was tightened by Viola [32].

We first state the results of Muroga and Viola required to formalize this construction.

Proposition 2.5 ([22, 23])**.** *Consider a threshold function $f : \{0,1\}^n \to \{0,1\}$ of the form $f(x) = \mathbb{1}\{a_1x_1 + \cdots + a_nx_n \leq b\}$. Then f is equivalent to another threshold function $f'(x) = \mathbb{1}\{a'_1x_1 + \cdots + a'_nx_n \leq b'\}$ where each $a'_1, \ldots a'_n, b'$ is at most $2^{-n}(n+1)^{(n+1)/2}$ in magnitude.*

In particular, Proposition 2.5 implies that we may assume in a $\mathsf{Th}(1)$ proof that, up to a factor of two in the size, every derived inequality has coefficients of magnitude at most $O(n^n)$.

Proposition 2.6 ([32])**.** *Suppose that each player $p \in [k]$ receives an input $x^{(p)} \in [-2^n, 2^n]$. Then there is a randomized number-in-hand protocol with error at most ε that determines whether $\sum_p x^{(p)} > s$ and communicates $O(k \log k \log(n/\varepsilon))$ bits.*

Corollary 2.7. *Suppose that each player $p \in [k]$ receives an input $x^{(p)} \in [2^t]$. Then they can execute a randomized number-in-hand protocol to determine whether $\sum_p a_p x^{(p)} \leq b$, where each $|a_p| \leq 2^w$ with error at most ε using at most $O(k \log k \log((w+t)/\varepsilon))$ bits of communication.*

Proof of Lemma 2.4. Given a threshold decision tree of depth d, we simply traverse it from the root. For each inequality, if the magnitudes of its weights are not bounded by $2^{O(n \log n)}$, then we replace it with an equivalent threshold function whose weights are bounded using Proposition 2.5. Then, using Corollary 2.7, the players communicate $O(k \log((n \log n)/\varepsilon) \log k)$ bits to compute the threshold function with error probability ε. Setting $\varepsilon = \Theta(1/d)$ and continuing in this manner, by a union bound the threshold functions are all computed correctly with constant probability.

Using the assumption that $d \leq n$, this yields a protocol communicating $O(dk \log k \log n)$ bits. □

Given a system of unsatisfiable inequalities $Ax \leq b$, and a partition of an assignment $\alpha \in \{0,1\}^n$ between k players, there is a natural (number-in-hand) communication game wherein players must communicate to a find an axiom violated by α. Lemma 2.4 implies the following result connecting communication complexity and proof complexity.

Lemma 2.8. *For any partition of n variables into k parts, if $\mathsf{Search}^k_{NIH}(Ax \leq b)$ requires t bits of communication, then any tree-like $\mathsf{Th}(1)$ refutation of $Ax \leq b$ requires size $2^{\Omega(t/(k \log k \log n))}$.*

Proof. By Proposition 2.2, given a size S tree-like $\mathsf{Th}(1)$ proof, we get a depth $d = O(\log S)$ threshold decision tree finding a violated axiom. By Lemma 2.4, there exists a communication protocol finding a violated axiom using $O(\log S\, k \log k\ \log n)$ bits of communication. This implies that $\log(S) k \log k \log n \geq ct$ for constant c, which in turn implies S is $2^{\Omega(t/(k \log k\ \log n))}$, as desired. □

2.3 Triangle-DAGs

Given a bipartite domain $X \times Y$, a *triangle* $T \subseteq X \times Y$ is any set that can be written as $T = \{(x,y) : a_T(x) < b_T(y)\}$ for some labelling $a_T : X \to \mathbb{R}$ and $b_T : Y \to \mathbb{R}$.

A triangle-DAG computing a search problem $\mathsf{Search} \subseteq (X \times Y \times \mathcal{O})$ is a directed acyclic graph D of fan-out at most 2 where each node $u \in D$ is associated with a triangle $T_u \subseteq X \times Y$ satisfying the following:

- there is a distinguished root node r with fan-in zero, and $T_r = X \times Y$, and
- for each non-sink node u with children v, v' we have $T_u \subseteq T_v \cup T_{v'}$, and

- each sink node u is labeled with an output o such that $T_u \subseteq \mathsf{Search}^{-1}(o)$.

Given these definitions, we restate Proposition 1.6 and include its proof for completeness.

Proposition 1.6. *[30, 16] Given a semantic cutting planes refutation for CNF formula φ of size s and any partition of the variables of φ into two sets, there is a size s triangle-DAG computing* Search_φ.

Proof. Let D be the DAG associated with the semantic CP refutation of φ. As we usually do, we orient the edges so that the last line of the proof, $0 \geq 1$, is the root of D. Each node $u \in D$ is associated with some linear inequality of the form $f(x) + g(y) \geq c$. If we then let $a_{T_h}(x) = f(x) - c$ and $b_{T_h}(y) = -g(y)$, then we have that $(x, y) \in T_u$ iff $f(x) + g(y) < c$, i.e iff (x, y) does not satisfy the inequality associated with u.

By definition, the root of D is associated with $X \times Y$. Since the inequalities corresponding to the children v and v' of node u imply the inequality associated with u, at least one of the children must not be satisfied by (x, y), i.e. we have that $T_u \subseteq T_v \cup T_{v'}$. Finally, each leaf $\ell \in D$ is associated with a clause o of φ that is falsified by (x, y), so $T_\ell \subseteq \mathsf{Search}_\varphi^{-1}(o)$, as desired. □

2.4 Min-Entropy and Deficiency

Definition 2.9. *For a distribution X defined on space Ω, define its* min-entropy *to be $H_\infty(X) := \min_\alpha \log \frac{1}{\mathbf{Pr}[X = \alpha]}$. Conversely, define the* deficiency *of X to be $D_\infty(X) := \log |\Omega| - H_\infty(X)$. For a rectangle $R = X_1 \times \ldots \times X_k \subseteq \Omega_1 \times \cdots \times \Omega_k$, define its deficiency to be $\sum_{i \in [k]} D_\infty(X_i)$.*

3 Multiparty Communication Lower Bounds for $m \leq n + \sqrt{n}$

In this section we prove Theorem 1.1, which we now recall.

Theorem 1.1. *The randomized number-in-hand communication complexity of $\mathsf{Coll}^k_{m,n^{1/k}}$ is $\Omega(n^{1-1/k} \log k \,/ 2^k)$ whenever $n + 1 \leq m \leq n + 2^{k-2} n^{1/k}$.*

The idea for the proof is to exhibit a random reduction from the decision problem $\mathsf{DISJ}^k_{m^{k-1}}$ to the collision problem. This is analogous to the approach of Itsykson and Riazanov [19] for number-on-forehead communication complexity which used lower bounds for disjointness in that model and a randomized decision-to-search reduction paradigm introduced by Raz and Wigderson [28] to prove lower bounds on the monotone depth complexity of matching. The details and parameters of our reduction are necessarily quite different.

In our setting, we embed k players' inputs to a disjointness problem into k matrices such that, when these matrices are concatenated, the resulting matrix has distinct rows if and only if the players' inputs were disjoint. We can then add a few "fake" rows to this matrix and run our algorithm for $\mathsf{Coll}^k_{m,n^{1/k}}$, and see if the collisions it finds involve the fake rows or not. If so, we conclude that the inputs were disjoint and if not, we know that they were not disjoint.

The following key combinatorial lemma allows us to carry out the first step of this process.

Lemma 3.1. *For all integers $k \geq 1$ there exist matrices $M^1_k \in \{0,1\}^{2^k \times k}$ and $M^0_k \in \{0,1\}^{2^k \times k}$ such that*

1. *M^1_k has 2^{k-1} unique pairs of identical rows.*
2. *For any string $b \in \{0,1\}^k$, define the matrix $M_k(b_1, \ldots b_k)$ as the matrix formed by making its i-th column equal to the i-th column of M^0_k if $b_i = 0$, and equal to the i-th column of M^1_k if $b_i = 1$. Then $M_k(b)$ has unique rows for all $b \neq \vec{1}$.*

We defer the proof of Lemma 3.1 to Section 3.1.

Proof of Theorem 1.1 using Lemma 3.1. As alluded to, we will reduce the NIH disjointness problem to our bit-pigeonhole problem. Namely, we will reduce $\mathsf{DISJ}^k_{m^{k-1}}$ to a $\mathsf{Coll}^k_{\widetilde{m},\widetilde{\ell}}$ communication game, where $\widetilde{m} = m^k 2^k + 2^{k-1}m$, and $\widetilde{\ell} = 2m$.

The players get $x^{(1)}, \ldots, x^{(k)} \subseteq [m^{k-1}]$ (viewed as bit strings of length m^{k-1}), and need to determine whether $x^{(1)} \cap \cdots \cap x^{(k)} = \varnothing$.

First, we define

$$B_{m^k}(j) := \begin{bmatrix} \mathrm{bin}_{m^k}(j) \\ \vdots \\ \mathrm{bin}_{m^k}(j) \end{bmatrix} \in \{0,1\}^{2^k \times k \log m},$$

where we have repeated the same row 2^k times.

For each $i \in [m^{k-1}]$, consider the matrix $M_k(x_i^{(1)}, \ldots, x_i^{(k)})$ from Lemma 3.1. Note that each player p knows the p-th column of $M_k(x_i^{(1)}, \ldots, x_i^{(k)})$ for all i, but without communication the other players do not know this column.

Then we can define

$$\widetilde{M} := \begin{bmatrix} M_k(x_1^{(1)}, \ldots, x_1^{(k)}) & B_{m^k}(0) \\ M_k(x_1^{(1)}, \ldots, x_1^{(k)}) & B_{m^k}(1) \\ \vdots & \vdots \\ M_k(x_1^{(1)}, \ldots, x_1^{(k)}) & B_{m^k}(m-1) \\ M_k(x_2^{(1)}, \ldots, x_2^{(k)}) & B_{m^k}(m) \\ M_k(x_2^{(1)}, \ldots, x_2^{(k)}) & B_{m^k}(m+1) \\ \vdots & \vdots \\ M_k(x_2^{(1)}, \ldots, x_2^{(k)}) & B_{m^k}(2m-1) \\ \vdots & \vdots \\ \vdots & \vdots \\ M_k(x_{m^{k-1}}^{(1)}, \ldots, x_{m^{k-1}}^{(k)}) & B_{m^k}(m^k-1) \end{bmatrix} \in \{0,1\}^{m^k \cdot 2^k \times (k \log m + k)}.$$

Observe that each player p can construct their "part" of this matrix without communicating by constructing the p-th column of every $M_k^{S_i}$ matrix (which only depends on $x^{(p)}$), and then taking the the p-th part of each of the $B_{m^k}(j)$ matrices.

Lemma 3.1 lets us connect the distinctness property of this matrix with the disjointness property of the players' inputs.

Claim 3.2. *If $(x^{(1)}, \ldots, x^{(k)})$ are disjoint, then $\widetilde{M}$ has distinct rows.*

Proof. The only possible collisions happen in every group of 2^k rows, since $B_{m^k}(j)$ has every row different from $B_{m^k}(i)$ for all $i \neq j$. Within these groups, by Lemma 3.1, if the inputs are disjoint then there are no collisions. □

Claim 3.3. *If X is not disjoint, then there are at least $2^{k-1}m$ pairs of colliding rows in $\widetilde{M}$.*

Proof. Any coordinate i for which $x_i^{(p)} = 1$ for all p generates $S_i = \varnothing$, which by Lemma 3.1 generates $2^{k-1}m$ such pairs, since input i was repeated m times. □

We cannot run any collision protocol for $\widetilde{M}$ yet, as there are not guaranteed collisions. To address this, the players use shared randomness to put an additional $2^{k-1}m$ rows at the bottom of $\widetilde{M}$. These rows will be chosen randomly with the following two properties:

1. Each fake row will be distinct.

2. Each player's "part" of the matrix (which consists of $\log m + 1$ columns) when restricted to these rows will repeat the $2m$ unique possible bit strings an addition $2^{k-1}m/(2m) = 2^{k-2}$ times.[8]

Denote this new matrix M. M now has "fake" collisions which involve any of the last $2^{k-1}m$ rows.

Let $\mathcal{A}$ denote the randomized protocol solving the $\mathsf{Coll}^k_{(2m)^k+2^{k-1}m,2m}$ problem.

Observe that if the inputs are disjoint, then the only collisions in M involve fake rows. The players would like to feed their parts of this matrix into $\mathcal{A}$ and conclude that their inputs are disjoint if the output involves a fake row, and conclude that they were not disjoint if the output involves two non-fake rows. However, this is problematic, as we have no guarantees over how $\mathcal{A}$ behaves, and it could always find a collision involving one of the last $2m$ rows (which it knows are fake), regardless of if there are other collisions.

This necessitates the following random shuffling.

1. Each player applies (the same) random permutation $\pi : [2^k m^k + 2^{k-1}m] \to [2^k m^k + 2^{k-1}m]$ which shuffles the rows of M.

2. Each player applies an individual random permutation $\pi^{(p)} : [2m] \to [2m]$ to each of their rows. Note that this preserves collisions/distinctness in the concatenation.

Denote $\vec{\pi} := (\pi, \pi^{(1)}, \ldots, \pi^{(k)})$, and call this final matrix $M_{\vec{\pi}}$.

Algorithm: The algorithm for disjointness is as follows: the players use their inputs and shared randomness to compute (without communication) their respective parts of 5 independent copies of an $M_{\vec{\pi}}$ constructed in the above manner, and run $\mathcal{A}$ using these as inputs. The players then examine the outputs $(i_1, j_1), \ldots (i_5, j_5)$ of the algorithm on these five inputs. They then exchange an additional $O(k \log m)$ bits to determine if each claimed collision actually was a collision. Finally, if any of the claimed collisions were actually collisions on rows that were not fake (under the appropriate permutation), then the players can conclude with certainty that their inputs were not disjoint. Otherwise, if $\mathcal{A}$ only ever finds colliding pairs that involve a row the players know is fake (or otherwise fail to find any collisions), then players guess that $(x^{(1)}, \ldots, x^{(k)})$ were disjoint.

Analysis: We analyze one iteration of the algorithm. Suppose $\mathcal{A}$ has error probability at most $1/3$ – that is, with probability at least $2/3$, it outputs two rows i, j that are equal.

Suppose $(x^{(1)}, \ldots, x^{(k)})$ are disjoint. Then by assumption, $\mathcal{A}$ finds a collision with probability at least $2/3$, and we know this collision will always involve a fake row by Claim 3.2. Therefore the players will correctly output that their inputs were distinct with probability at least $2/3$, and this is only improved by the five-fold repetition.

Otherwise, suppose that the players' inputs were not disjoint. Suppose further that $\mathcal{A}$ successfully finds a collision – this happens with probability at least $2/3$. Recall that by Claim 3.3 $M_{\vec{\pi}}$ will have at least $2^{k-1}m$ distinct pairs of real collisions. Adding the $2^{k-1}m$ fake rows produced additional "fake" collisions. These fake rows could have created up to $2^{k-1}m$ additional unique pairs of fake collisions, or could have "joined" the real collisions, creating up to $2^{k-1}m$ groups of 3 equal rows in $M_{\vec{\pi}}$.

If $\mathcal{A}$ outputs a collision from one of the groups of three, then because we applied random permutations to the rows, it is equally likely to have chosen any of the 3 possible pairs. Therefore, with probability at least $1/3$, it outputs a real collision, and the players successfully discover that they are not disjoint. Otherwise, if $\mathcal{A}$ outputs one of the unique collision pairs, then (again because we have applied random permutations to the rows), any such unique collision is equally likely to be output. If t of the fake rows formed a group of three with real collisions, then that leaves at most $2^{k-1}m - t$ fake rows to collide with a different unique row. It also leaves $2^{k-1}m - t$ untouched real collisions, so $\mathcal{A}$ outputs a real collision with probability at least $1/2$. Either way, the probability that $\mathcal{A}$ outputs a real collision is at least $2/3 \cdot 1/3 = 2/9$.

Therefore, after repeating this 5 times independently, the probability of seeing at least one real collision is at least $1 - (7/9)^5 > 2/3$.

[8]This is important because if we bias and have a certain fake row appear more often in the input for player p, then $\mathcal{A}$ could potentially detect and use this to its advantage.

Let $n := (2m)^k$. We have shown that if $\mathsf{Coll}^k_{\widetilde{m},\widetilde{\ell}}$ with input size $\widetilde{m} = 2^k m^k + 2^{k-1} m = n + 2^{k-2} n^{1/k}$ can be solved with $o(n^{1-1/k} \log k / 2^k)$ communication, then we can solve the decision disjointness problem with $o(n^{1-1/k} \log k / 2^k) + O(k \log m)$ which is at most $o(m^{k-1} \log k)$, contradicting the $\Omega(m^{k-1} \log k)$ lower bound from Theorem 2.3. □

3.1 Proof of Lemma 3.1

We first recall Lemma 3.1.

Lemma 3.1. *For all integers $k \geq 1$ there exist matrices $M_k^1 \in \{0,1\}^{2^k \times k}$ and $M_k^0 \in \{0,1\}^{2^k \times k}$ such that*

1. *M_k^1 has 2^{k-1} unique pairs of identical rows.*
2. *For any string $b \in \{0,1\}^k$, define the matrix $M_k(b_1, \ldots b_k)$ as the matrix formed by making its i-th column equal to the i-th column of M_k^0 if $b_i = 0$, and equal to the i-th column of M_k^1 if $b_i = 1$. Then $M_k(b)$ has unique rows for all $b \neq \vec{1}$.*

Proof. Let $\mathcal{E}_k \subseteq \{0, \ldots, k-1\}$ be the set of integers with an even number of 1s in their binary representation. Define

$$M_k^1 = \begin{bmatrix} \mathrm{bin}_k(y_1) \\ \mathrm{bin}_k(y_1) \\ \mathrm{bin}_k(y_2) \\ \mathrm{bin}_k(y_2) \\ \vdots \\ \mathrm{bin}_k(y_{2^{k-1}}) \\ \mathrm{bin}_k(y_{2^{k-1}}) \end{bmatrix} \in \{0,1\}^{2^k \times k},$$

where y_i is the i-th smallest number in $\mathcal{E}_k$.

We pause here to note that each of the columns of M_k^1 are actually each the truth table of a linear function. Let $f_1 : \{0,1\}^k \to \{0,1\}$ be the linear function $f_1(x) = \langle x, e_1 \rangle$, where e_1 is the first standard basis vector. Then we can describe the first column of M_k^1 as the truth table of f_1. More generally, we have that for $i < k$ the i-th column of M_k^1 is the truth table of $f_i(x) = \langle x, e_i \rangle$, and the last column is the truth table of $f_k(x) = \langle x, e_1 + \cdots + e_{k-1} \rangle$.

If we define F_k^1 to be the matrix whose column i is the vector whose inner product we are taking with x in f_i, and $B_k \in \mathbb{F}_2^{2^k \times k}$ whose rows are the binary strings written in order, then we have that

$$M_k^1 = \begin{bmatrix} \mathrm{bin}_k(0) \\ \mathrm{bin}_k(1) \\ \vdots \\ \mathrm{bin}_k(2^k - 1) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \cdots & 1 \\ 0 & 1 & 0 & \cdots & 1 \\ 0 & 0 & 1 & \ddots & 1 \\ & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} =: B_k F_k^1$$

where all operations are over $\mathbb{F}_2$. M_k^1 has repeated rows precisely because F_k^1 has linearly dependent columns.

With this perspective in mind, if we can find a $k \times k$ matrix F_k^0 over $\mathbb{F}_2$ such that replacing any (nonzero) number of columns of F_k^1 with corresponding columns in F_k^0 produces a matrix with linearly independent columns, then we are done, as we can let $M_k^0 := B_k F_k^0$.

We define F_k^0 to be the following lower triangular matrix:

$$F_k^0 := \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix},$$

Clearly F_k^0 is full rank.

We claim that replacing any set nonempty set S of columns of F_k^1 with the corresponding columns in F_k^0 produces a matrix F_k^S with linearly independent columns. Consider arbitrary k, and arbitrary nonempty $S \subseteq [k]$.

Case 1: Suppose $k \in S$, that is, the last column of F_k^S is e_k. Then our matrix is lower triangular with 1s on the diagonal, and so has linearly independent columns.

Case 2: Suppose $k \notin S$, that is, the last column of F_k^S is $e_1 + \cdots + e_{k-1}$. In this case, the first $k-1$ columns are lower triangular with 1s on the diagonal, and therefore their span is equal to $\mathsf{span}\{e_1, \ldots, e_{k-1}\}$. It suffices then to show that also e_k is in the column span. Towards this goal, take the minimal $0 < i < k$ such that $i \in S$, and observe that summing the first i columns of F_k^S equals $\vec{1}$, the all 1s vector. Adding this to the last column produces e_k, so we are done. □

4 Tree-like Proof Complexity Lower Bounds

In this section, we prove the following more detailed version of Theorem 1.2.

Theorem 4.1. *When $m \leq n + 2^{k-2}n^{1/k}$, any tree-like semantic cuttings planes refutation of* BPHP_n^m *must have size at least* $2^{\Omega(n^{1-1/k}2^{-k}/(k \log n))}$.

Proof. The theorem follows quite readily from Lemma 2.8 and the fact that $\mathsf{Search}^k_{\mathrm{BPHP}_n^m}$ reduces to $\mathsf{Coll}^k_{m,n^{1/k}}$. If we translate BPHP_n^m to a system of inequalities, then there are $O(nm^2) = O(n^3)$ inequalities on $m \log n$ variables.

Lemma 2.8 then says that any tree-like semantic cutting planes proof of the unsatisfiability of BPHP_n^m must have size at least $2^{\Omega(t/(k \log k \log n))}$. By Theorem 1.1, $t = \Omega(n^{1-1/k}2^{-k} \log k)$. Plugging this in yields that the tree-like size must be at least $2^{\Omega(n^{1-1/k}2^{-k}/(k \log n))}$. □

From Theorem 4.1 we achieve Theorem 1.2, which we restate now.

Corollary 4.2. *Any tree-like semantic cutting planes refutation of* BPHP_n^m *requires size* $2^{n^{1-2/\sqrt{\log n}-o(1/\sqrt{\log n})}}$ *when* $m \leq n + 2^{2\sqrt{\log n}-2}$.

Proof. Let $k = \sqrt{\log n}$. Plugging this into the bound from Theorem 4.1, we get that $m \leq n + 2^{\sqrt{\log n}-2}n^{1/\sqrt{\log n}} = 2^{2\sqrt{\log n}-2}$. In this regime, Theorem 4.1 gives the size lower bound

$$2^{\Omega(n^{1-1/\sqrt{\log n}}n^{-1/\sqrt{\log n}}/(\log^{3/2} n))} = 2^{cn^{1-2/\sqrt{\log n}-1.5 \log\log n/\log n}},$$

for appropriate constant c. Using the fact that $c = n^{\log c/\log n}$, the bound becomes

$$2^{n^{1-2/\sqrt{\log n}-o(1/\sqrt{\log n})}}.$$ □

5 Triangle-DAG Lower Bounds

In this section, we prove Theorem 1.7, which we restate here for convenience.

Theorem 1.7. *For the natural partition of variables $V_x \sqcup V_y = [m \log n]$, any triangle-DAG computing* $\mathsf{Search}_{\mathrm{BPHP}_n^m}$ *requires size* $2^{n^{1/4}/\sqrt{2}-2}$.

Before proving Theorem 1.7, we need the following notion of triangle slices. For a triangle $T \subseteq X \times Y$ and a given $x \in X$, we define the slice T^x to be $T \cap (\{x\} \times Y)$, and define the slice T^y analogously. Note that for any triangle T defined by functions a and b there is a natural total pre-order on X where $x \preceq x'$ iff $a(x') \leq a(x)$ and hence $T^x \subseteq T^{x'}$; similarly, there is an total pre-order on Y where $y \preceq y'$ iff $b(y) \leq b(y')$ and hence $T^y \subseteq T^{y'}$. This immediately implies the following.

Proposition 5.1. *Given a triangle T, if $y \in Y$ is such that $|T^y|$ is maximized, then for all $(x, y') \in T$ we have $(x, y) \in T$.*

We will have the following proposition.

Proposition 5.2. *For any triangle T and rectangle R, $T \cap R$ is a triangle.*

Proof. By definition, $T = \{(x, y) : a(x) < b(y)\}$ for some a, b. If $R = X' \times Y'$, simply modify $a(x) = \infty$ for all $x \notin X'$ and $b(y) = -\infty$ for all $y \notin Y'$. □

The proof of Theorem 1.7 uses the *bottleneck counting* method of Haken and Cook for cutting planes proofs [14] as adapted for direct use on triangle-DAGs by Sokolov [31].

Let $V(D)$ denote the nodes of a triangle-DAG D computing $\mathsf{Search}_{\mathrm{BPHP}_n^m}$ under the natural partition of variables. At a high level, to prove Theorem 1.7, we will construct a partial function $\mu : X \cup Y \to V(D)$ such that

(a) (Lemma 5.5) most $x \in X$ and $y \in Y$ are mapped by μ to a node of D, and

(b) (Lemma 5.6) there is no node of D with too many x or y assigned to it by μ.

It will then be trivial to combine these two properties and conclude that $|V(D)|$ must be large[9].

For any $i \neq j \in [m]$, there is an associated rectangle $R_{i,j} \subseteq X \times Y$ of inputs that violate at least one of the n clauses enforcing an inequality between coordinates i and j given by

$$R_{i,j} = \{(x, y) \in X \times Y \mid x_i = x_j,\ y_i = y_j\}.$$

Observe that the set of all rectangles $\{R_{i,j} : i \neq j \in [m]\}$ covers the entire space $X \times Y$ as long as $m > n$. (That is, $X \times Y = \bigcup_{i \neq j \in [m]} R_{i,j}$.) Note that in order for a triangle-DAG to correctly compute $\mathsf{Search}_{\mathrm{BPHP}_n^m}$, it must, at the very least, have each sink be contained in a single $R_{i,j}$.

Definition 5.3. *For a node $u \in V(D)$, let T_u denote the triangle associated with that node. Define the* width *$w(T, z)$ of z in triangle T to be the minimal size of any subset $S \subseteq \{R_{i,j} : i \neq j \in [m]\}$ such that $T^z \subseteq \bigcup_{R \in S} R$.*

With this definition in hand, we can define our $\mu : X \cup Y \to V(D)$ using Algorithm 1 which depends on a parameter k that we will eventually set to $\sqrt{n}/4$.

Intuitively, in using Algorithm 1 we are going backwards through our DAG D, and whenever the protocol had too many options for what clause to output when the protocol was at a node u for a fixed $z \in X \cup Y$

[9]The name bottleneck counting came from its original version due to Haken [13] in which a class of *full* input assignments was mapped by a function μ to nodes in a proof DAG and each input assignment could be viewed as 'flowing' from the root to a particular sink node (defined by the assignment). In that case, μ identified a node on such a path, a bottleneck node, that did not permit many assignments to flow through it. Since there were many assignments, only few of which could pass through any bottleneck node identified by μ, there must be many bottleneck nodes in the DAG. In the modified form for cutting planes/triangle-DAGs, while the full assignment is flowing through the triangle-DAG, the mapping μ is based on half of the input assignment, either the x or y portion, and the sink node is not unique; not all assignments are assigned in this way, but many are.

Algorithm 1: Defining the partial map $\mu : X \cup Y \to V(D)$

Input: triangle-DAG D on input set $X \times Y$ with node set $V(D)$
Output: a partial map $\mu : X \cup Y \to V(D)$

Let $X' := X$ and $Y' := Y$ be the remaining coordinates we have not yet assigned by μ
for *u in some topological ordering of $V(D)$, starting from the sinks* **do**
  $\widetilde{T}_u := T_u \cap (X' \times Y')$
  for $x \in X'$ **do**
    if $w(\widetilde{T}_u, x) \geq k$ **then**
      Let $\mu(x) := u$ and delete x from X'.
    $\widetilde{T}_u := T_u \cap (X' \times Y')$
  for $y \in Y'$ **do**
    if $w(\widetilde{T}_u, y) \geq k$ **then**
      Let $\mu(y) := u$ and delete y from Y'.
    $\widetilde{T}_u := T_u \cap (X' \times Y')$
return μ

(which corresponds to large width), we assign z to u. Arguing that $|\mu^{-1}(u)|$ is not too large is thus arguing that a protocol cannot make huge strides in learning about which clause to output for too many $z \in X \cup Y$ suddenly in one single node u.

We now make some basic observations about our definition of μ. First, note that no $z \in X \cup Y$ is assigned to any sink node of D, since every sink node $s \in V(D)$ is wholly contained in an $R_{i,j}$, and therefore we have that $w(T_s, z) = 1$ for all z. The following observation will be of critical importance in our analysis.

Claim 5.4. *During the execution of Algorithm 1, for every u in triangle-DAG D and every $z \in X' \cup Y'$, $w(\widetilde{T}_u, z) \leq 2k$.*

Proof of Claim. By definition in a triangle-DAG, if v and v' are children of node u, then we have that $T_u \subseteq T_v \cup T_{v'}$. In particular, we know that if $z \in X \cup Y$ was not assigned to nodes v or v', then $w(\widetilde{T}_u, z) \leq 2k$, since we can take the union of the coverings for $\widetilde{T}^z_v$ and $\widetilde{T}^z_{v'}$. □

Finally, we note that at the end of Algorithm 1 at the root r, for all $z \in X' \cup Y'$, we have that $w(\widetilde{T}_r, z) < k$.

Lemma 5.5. *Let $k \leq \sqrt{n}/2$. For μ constructed using Algorithm 1, at least $|X|/2$ elements of X are assigned by μ or at least $|Y|/2$ elements of Y are assigned by μ.*

Proof. At the end of Algorithm 1 after processing the root r of D, since $T_r = X \times Y$, we have $\widetilde{T}_r = X' \times Y'$ which is the rectangle of unassigned inputs. If $|X'| \leq |X|/2$ then we are done; otherwise, let $x \in X'$. Since $w(\widetilde{T}_r, x) < k$, there exists a set S, $|S| < k$, such that for all $y \in Y'$, $y_i = y_j$ for at least one $(i, j) \in S$.

So, we obtain

$$\mathop{\mathbf{Pr}}_{y \sim Y}[y \in Y'] \leq \mathbf{Pr}[\exists (i,j) \in S \text{ s.t. } y_i = y_j] \leq |S| \max_{(i,j) \in S} \mathbf{Pr}[y_i = y_j] < k/\sqrt{n} \leq 1/2.$$

Hence, $|Y'| < |Y|/2$, as desired. □

This shows that a large number of elements of $X \cup Y$ are assigned by μ. It remains to show that each node in the protocol D only has a bounded number of elements of $X \cup Y$ assigned by μ.

Lemma 5.6. *Let μ be defined by Algorithm 1 with $k = \sqrt{n}/4$. For all all $u \in V(D)$, μ maps at most $|X| \cdot 2^{-n^{1/4}/\sqrt{2}+1}$ elements of X to u; the analogous bound also holds for Y.*

Algorithm 2: Defining tree $\mathcal{T}$, which encodes a set of potential coverings of $\widetilde{T}^x$ for all $x \in X$.

Input: A triangle $\widetilde{T} \subseteq X \times Y$ such that $w(\widetilde{T}, y) \leq 2k$ for all $y \in Y$
Output: A tree $\mathcal{T}$ of potential coverings of $\widetilde{T}^x$ for all $x \in X$.

Initialize $\mathcal{T}$ as a tree with a single node labelled $\widetilde{T}$
Let Leaves $:= \{\widetilde{T}\}$
while Leaves $\neq \varnothing$ **do**
  Choose some T from Leaves, and remove it from Leaves
  Choose a $y \in Y$ such that $|T^y|$ is maximized
    `/* For all` $x \in X$ `if` $T^x \neq \varnothing$ `then` $(x, y) \in T$ `by Proposition 5.1. */`
  Let $\{R_{i_1,j_1}, \ldots, R_{i_\ell,j_\ell}\} \subseteq \{R_{i,j} : i \neq j \in [m]\}$ be a minimal covering for T^y.
    `/* Note that` $\ell \leq 2k$`; see Fig. 1 for a visualization of this step. */`
  for $m = 1, \ldots, \ell$ **do**
    Let $R := R_{i_m,j_m} = X_R \times Y_R$
    Let $T_R := T \cap (X_R \times (Y \setminus Y_R))$
      `/* This is still a triangle by Proposition 5.2. */`
    Add edge to $\mathcal{T}$ labelled by (i_m, j_m) from the leaf of $\mathcal{T}$ labelled T to new node labelled by T_R
    if $T_R \neq \varnothing$ **then**
      Add T_R to Leaves.
return $\mathcal{T}$

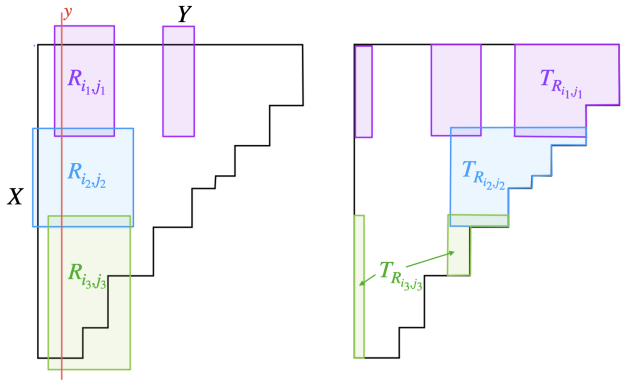

Figure 1: A single iteration of Algorithm 2.

Proof. We fix any $u \in V(D)$ and focus on the number of elements $x \in X$ that μ maps to u; the bound for the number of such $y \in Y$ is identical. In particular, consider the values of X' and Y' and $\widetilde{T}_u$ in Algorithm 1 immediately before processing node u; any $x \in X$ that μ maps to u must be in X'.

Our strategy will be to construct a set of *potential* coverings for the slices $\{\widetilde{T}_u^x : x \in X'\}$ by monochromatic rectangles. We will then aim to show that most $x \in X'$ have width $< k$ even when restricted to this set of potential coverings, thereby limiting how many $x \in X'$ could be assigned by μ to u.

We now describe this potential covering in detail. Intuitively, our strategy will be to construct a tree $\mathcal{T}$, each of whose nodes is labelled by a triangle of inputs that are not yet covered, where each edge of the tree is labelled with a pair $i \neq j \in [m]$. We say that $x \in X'$ is *consistent* with a node $t \in \mathcal{T}$ iff for all (i, j) labels on the edges of the unique path to t in $\mathcal{T}$, we have that $x_i = x_j$. Furthermore, $\mathcal{T}$ will be constructed so that if $x \in X'$ is consistent with a leaf t of $\mathcal{T}$, then $\widetilde{T}_u^x$ is covered by $\{R_{i,j} : (i, j) \text{ is on the path to } t\}$. We produce the tree $\mathcal{T}$ of potential coverings of of $\widetilde{T}_u^x$ for all $x \in X'$ using Algorithm 2 applied with $\widetilde{T} = \widetilde{T}_u$, $X = X'$, and $Y = Y'$ which satisfy the preconditions by Claim 5.4.

First, observe that any $x \in X'$ such that $\widetilde{T}^x \neq \varnothing$ is consistent with at least one leaf in $\mathcal{T}$, by the choice of y

in Algorithm 2 along with Proposition 5.1. We define the set of equality constraints of a leaf $t \in \mathcal{T}$ to be the set $E(t) := \{i \neq j \in [m] : (i,j) \text{ is on the unique path to } t \text{ in } \mathcal{T}\}$. Observe that we have the desired property that if x is consistent with a leaf t of $\mathcal{T}$, then $\widetilde{T}_u^x$ is covered by $\{R_{i,j} : (i,j) \in E(t)\}$, thereby ensuring that $w(\widetilde{T}_u, x) \leq \text{depth}_{\mathcal{T}}(t)$.

Therefore, if μ maps $x \in X'$ to u then x must be consistent with a leaf of $\mathcal{T}$ of depth at least k. Let $\mathcal{T}'$ denote tree $\mathcal{T}$ truncated at depth k. Thus, the total number of $x \in X'$ that μ maps to u is at most

$$\sum_{\substack{t \in \mathsf{Leaves}(\mathcal{T}') \\ \text{depth}_{\mathcal{T}'}(t) = k}} |\{x \in X : x \text{ is consistent with } t\}|. \tag{1}$$

Remark 5.7. Before explaining the details of our analysis, it is useful to discuss a naive bound on (1) that does not suffice to yield our bounds: Naively, there could be $(2k)^k$ leaves in $\mathcal{T}'$ at depth k, and each such leaf t could have the property that $E(t)$ forms a clique on $\ell = O(\sqrt{k})$ coordinates when viewed as a graph. In this case, the fraction of $x \in X$ that could be consistent with the intersection $R_{i_1,j_1} \cap \cdots \cap R_{i_k,j_k}$ is $(1/\sqrt{n})^\ell$. If we simply take a union bound over all $(2k)^k$ paths, the upper bound we would get on (1) is $(2k)^k \cdot (1/\sqrt{n})^\ell$ which, since ℓ is only $O(\sqrt{k})$, is larger than 1 unless $k^{\sqrt{k}}$ is at most polynomial in n, which requires that k is $o(\log^2 n)$.

In light of Remark 5.7, we need to be more careful about how we bound (1). For $t \in V(\mathcal{T}')$, we define its *equality graph* $G(t)$ to be a graph with vertex set $[m]$ and edge set $E(t)$. We use this name because if x is consistent with t, then $x_i = x_j$ for all $(i,j) \in E(t)$. We say that an edge (i,j) denoting $x_i = x_j$ is *implied* by equality graph $G(t)$ if adding it completes a cycle in $G(t)$.

The key observation is the following.

Claim 5.8. *Let t be a node in $\mathcal{T}'$, and $G(t)$ its associated equality graph. If t has out-degree strictly larger than 1 in $\mathcal{T}'$, then none of the outgoing edges can be labelled by an (i,j) that is implied by $G(t)$.*

Proof of Claim. We prove the contrapositive. Consider any x consistent with the path to t in $\mathcal{T}'$ (meaning $x_i = x_j$ for all $(i,j) \in E(t)$). Let T be the triangle labelling node t and consider the y that we choose when processing T in Algorithm 2. If there is some outgoing edge from t labelled (i,j) that is implied by $G(t)$, then $R_{i,j} \cap T^y \neq \varnothing$ and hence $y_i = y_j$. Since we know that $x_i = x_j$ is already implied in T, the single rectangle $R_{i,j}$ covers all of T^y, so t will have out-degree 1 in $\mathcal{T}$. □

Claim 5.8 implies that every path in $\mathcal{T}'$ goes through a node that branches at least $\sqrt{2k}$ times, since k edges must imply equality constraints on at least $\sqrt{2k}$ coordinates (the worst case is a clique). We simplify $\mathcal{T}'$ further by collapsing nodes of $\mathcal{T}'$ with out-degree 1. Then we are left with $\mathcal{T}''$, which is a tree with every path of length between $\sqrt{2k}$ and k. Since, the input distribution on X is uniform and none of the out-degree 1 edges of $\mathcal{T}'$ involve new constraints by Claim 5.8, the probability that x is consistent with any leaf $t \in \mathcal{T}''$ is precisely $(1/\sqrt{n})^{\text{depth}_{\mathcal{T}''}(t)}$. Then we can bound (1) as follows:

$$\begin{aligned}
&\sum_{\substack{t \in \mathcal{T}' \\ \text{depth}_{\mathcal{T}'}(t) = k}} |\{x \in X : x \text{ is consistent with } t\}| \\
&\quad = \sum_{t \in \mathsf{Leaves}(\mathcal{T}'')} |\{x \in X : x \text{ is consistent with } t\}| \\
&\quad = \sum_{i=\sqrt{2k}}^{k} \sum_{\substack{t \in \mathsf{Leaves}(\mathcal{T}'') \\ \text{depth}_{\mathcal{T}''}(t) = i}} |\{x \in X : x \text{ is consistent with } t\}| \\
&\quad \leq \sum_{i=\sqrt{2k}}^{k} (2k)^i (1/\sqrt{n})^i \qquad \text{since } \mathcal{T} \text{ and hence } \mathcal{T}'' \text{ has out-degree at most } 2k \text{ by construction}
\end{aligned}$$

$$\leq |X| \cdot \sum_{i=\sqrt{2k}}^{k} (1/2)^i \qquad \text{since } k = \sqrt{n}/4$$
$$\leq 2^{-\sqrt{2k}+1}|X|.$$

This implies that the number of $x \in X$ that get assigned to u by μ is at most $|X| \cdot 2^{-\sqrt{2k}+1} \leq |X|2^{-n^{1/4}/\sqrt{2}+1}$ as required. The analogous bound holds for the number of y assigned to u by μ. □

We now have all we need to prove our main theorem.

Proof of Theorem 1.5. Let $k = \sqrt{n}/4$. By Lemma 5.5, μ given by Algorithm 1 maps at least $1/2$ of the elements of x or elements of Y to vertices of D. Choose the applicable X or Y. By Lemma 5.6, at most a $2^{-n^{1/4}/\sqrt{2}+1}$ fraction of these can map to a single vertex of D. Therefore D has at least $2^{n^{1/4}/\sqrt{2}-2}$ vertices. □

Remark 5.9. Hrubeš and Pudlák [17] proved their lower bound using the interpolation method combined with a modified form of a general monotone switching lemma of Jukna [20]. The construction of Algorithm 2 has some flavor of the monotone switching lemma arguments, but the overall argument here seems fairly different. It would not be surprising that if one could modify this method to yield a similar lower bound to the one we give here, but exactly how to do that is not clear. In any case, we find the form of the bottleneck counting argument appealing.

6 Multiparty Communication Lower Bounds for all $m > n$

In this section, we prove:

Theorem 1.3. *For all $m > n$ and $2 \leq k \leq \log n\ /4$, the randomized number-in-hand communication complexity of $\mathsf{Coll}^k_{m,n^{1/k}}$ is $\Omega(n^{1/2-1/(2k)}/k)$.*

Before proving Theorem 1.3, we show that it is essentially optimal by a showing a simple protocol nearly matches our lower bound.

Proposition 6.1. *For $m \geq n + n^{1-1/k}$ and $k \leq \log n\ /4$, there is a randomized k-party number-in-hand protocol for $\mathsf{Coll}^k_{m,n^{1/k}}$ with complexity $O(n^{1/2-1/(2k)} \log n)$.*

Proof Sketch. Without loss of generality, $m = n + n^{1-1/k}$, since we can always ignore the remaining coordinates. By the pigeonhole principle there must exist a subset of at least $\ell = m/n^{1/k}$ coordinates on which Player 1's input is the same. The protocol is as follows. Player 1 first announces a random subset S of these coordinates of size $t = 16n^{1/2-1/(2k)}$. Players 2 through $k-1$ send vectors of length t giving their values on each element of S. Player k announces a collision among the pairs of elements of S, if there is one. Player 1's announcement requires $O(n^{1/2-1/(2k)} \log n)$ bits, while the remaining $k-1$ announcements require at most $O(n^{1/2-1/(2k)} \log(n^{1/k}))$ bits each. The total bound on the communication is therefore $O(n^{1/2-1/(2k)} \log n)$, as desired. It suffices to now argue that player k will have a collision to announce with good probability.

For any input x, it will be convenient to think of of a graph G_x whose vertices are coordinates $i \in [m]$, and where an edge (i, j) exists iff $x_i = x_j$. Observe that, for any input, G_x is a collection of non-trivial cliques, where the sum of the number of vertices in the cliques is at least $r = m - n$. This implies that there are at least $r/2$ vertex-disjoint edges in G_x.

When viewed in this way, Player 1's announcement is equivalent to them outputting a random clique on t vertices (out of ℓ possible vertices). In order for the players to succeed, it is sufficient for this clique to contain one of the $r/2$ vertex-disjoint edges.

Consider sampling our random clique by first sampling $t/2$ random vertices of the t vertices that Player 1 samples. Of our $r/2$ vertex-disjoint edges, in expectation $q = (r/2)(t/2)/\ell = rt/(4\ell)$ of them have a vertex we sampled in this first phase. By Hoeffding's bound on the tail of the hypergeometric distribution, with

probability at least $1 - 2^{-q/2}$ we see at least $q/4$ vertices. If any have both endpoints already sampled, then we are done, so assume this is not the case.

We then sample the remaining $t/2$ disjoint vertices to include in Player 1's clique. In order to succeed, one of these $t/2$ additional vertices must complete one of our $q/4$ edges. The probability that this does not happen is at most

$$(1 - \frac{t/2}{\ell})^{q/4} \leq e^{-tq/(8\ell)} = e^{-(t/\ell)^2 \cdot r/32}.$$

Now, plugging in $(t/\ell)^2 = (16n^{1/2+1/2k}/m)^2 = 16^2 n^{1+1/k}/m^2$, we see that since $m = n + n^{1-1/k}$, we have $r = n^{1-1/k}$ and hence, conditioned on finding sampling at least $q/4$ vertices for the first set, the probability of failure is at most

$$e^{-16^2 n^{1+1/k}/m^2 \cdot n^{1-1/k}/32} \leq e^{-8n^2/(2n)^2} \leq e^{-2}.$$

There the total failure probability is at most $2^{-q/2} + e^{-2} \leq 1/4$. □

For the proof of Theorem 1.3 we simplify and generalize ideas of Yang, Zhang, and Wang [34, 33] to the $k > 2$-party setting for $\mathsf{Coll}^k_{m,n^{1/k}}$.

Suppose that we have a randomized protocol Π solving $\mathsf{Coll}^k_{m,n^{1/k}}$. We follow [33] and decompose each rectangle in protocol tree further into "structured" subrectangles for which we need the following definition.

Definition 6.2 (Pseudorandomness/density)**.** *For a random variable X supported on Ω^J, we say it is $(1-\delta)$-*dense *if for all $I \subseteq J$, $H_\infty(X(I)) \geq (1-\delta)\ |I| \log |\Omega|$, where $X(I)$ denotes the random variable marginalized to the coordinates in I.*

For $\mathsf{Coll}^k_{m,n^{1/k}}$ we have $\mathrm{supp}(X_i) = \Omega^m$ for $\Omega = [n^{1/k}]$ and we will choose $\delta = \frac{1}{\log n}$.

Definition 6.3 (Structured rectangle)**.** *Given a rectangle $R = X_1 \times \ldots \times X_k$, where each $X_i \subseteq \Omega^m$, we say that R is* structured *iff there exist $J_1, \ldots, J_k \subseteq [m]$ and $\tau_1, \ldots, \tau_k$ where each $\tau_i : [m] \setminus J_i \to \Omega$ such that*

- *X_i is $(1-\delta)$-dense on the coordinates in J_i and*
- *X_i is fixed to τ_i on the coordinates in $\overline{J_i} := [m] \setminus J_i$.*

The high level idea in [34, 33] is to recursively partition each rectangle in the protocol into structured subrectangles so that, if the players have not communicated much, then the average number of fixed coordinates in a random leaf subrectangle is small.

Let us now detail this algorithm. It is similar to the algorithm in [34], but crucially different in that we do not keep track of error sets/subrectangles as we go.[10]

We shall need one important tool from [11], which decomposes an arbitrary random variable into structured subparts.

Lemma 6.4 (Density-restoring partition from [11])**.** *Given X on Ω^J, we can partition it as $X = \bigcup_i X^i$ such that*

1. *X^i has an associated I_i and τ_i so that for all $x \in X^i$, $x(I_i) = \tau_i$,*
2. *$X^i(J \setminus I_i)$ is $(1-\delta)$-dense, and*
3. *$D_\infty(X^i(J \setminus I_i) \leq D_\infty(X(J)) - \delta|I_i| \log |\Omega| + \log \frac{|X|}{|\cup_{j \geq i} X_j|}$.*

[10] Keeping track of the errors as we go seems to lead to cylinder intersections when $k > 2$ and more complicated analyses when $k = 2$. For that reason, our decomposition algorithm and analysis is more similar in spirit to what can be found in [33].

Algorithm 3: Decomposition/Sampling Process producing distribution $\mathcal{R}^{\text{leaf}}$ (as in [33])

Input: Deterministic k-party NIH protocol Π with inputs in $(\Omega^m)^k$
Output: A randomly sampled structured subrectangle R from a leaf node of Π

Let $V = X_V^1 \times \ldots \times X_V^k$ be the current rectangle of the protocol, which is initialized to the root.
Let $R = X_1 \times \ldots \times X_k$ be the current subrectangle, which is initialized to the entire root.
Initialize $J_i = [m]$ for all $i \in [k]$
while *v is not a leaf of* Π **do**
    Let i be the speaker at v, and let $X_v^i = X_{v_0}^i \cup X_{v_1}^i$ be induced by the communicated bit.
    Let $X_i^0 = X_{v_0}^i \cap X_i$ and $X_i^1 = X_{v_1}^i \cap X_i$
    w.p. $\frac{|X_i^0|}{|X_i|}$ set $v \leftarrow v_0$ and $X_i \leftarrow X_i^0$, and otherwise set $X_i \leftarrow X_i^1$ and $v \leftarrow v_1$
    if *X_i is not $(1-\delta)$-dense* **then**
        Decompose X_i using Lemma 6.4 into $X^1, \ldots, X^r$
        Set $X_i \leftarrow X^j$ and $J_i \leftarrow J_i \setminus I_j$ w.p. $\frac{|X^j|}{|X_i|}$
        Let $R \leftarrow X_1 \times \ldots \times X_k$
return R

Let $\mathcal{R}^{\text{leaf}}$ denote the distribution over subrectangles at the leaf level obtained via this sampling process. Note that each subrectangle is sampled uniformly according to its size. Each R has associated with it $J_1 = J_1(R), \ldots, J_k = J_k(R)$ and $\tau_1 = \tau_1(R), \ldots, \tau_k = \tau_k(R)$.

Implicit in the work of [33] (see the proof of their Lemma 3.2) is the following lemma.

Lemma 6.5. *Let Π be a deterministic k-party number-in-hand protocol solving any search problem over $(\Omega^m)^k$. For the distribution $\mathcal{R}^{\text{leaf}}$ given by Algorithm 3,*

$$\mathop{\mathbf{E}}_{R \sim \mathcal{R}^{\text{leaf}}} \sum_i |\overline{J_i}| \leq 2 \cdot CC(\Pi)/(\delta \log |\Omega|).$$

In particular, Lemma 6.5 yields the following corollary.

Corollary 6.6. *Let Π be a deterministic k-party number-in-hand protocol solving any search problem over $([n^{1/k}]^m)^k$. For the distribution $\mathcal{R}^{\text{leaf}}$ given by Algorithm 3 with $\delta = 1/\log n$,*

$$\mathop{\mathbf{Pr}}_{R \sim \mathcal{R}^{\text{leaf}}} \left[\sum_i |\overline{J_i}|^2 \geq 400 \ (k \cdot CC(\Pi))^2 \right] \leq 1/10.$$

Proof. We observe that in this case the expectation bound on $\sum_i |\overline{J_i}|$ in Lemma 6.5 is simply $2k \cdot CC(\Pi)$. Applying Markov's inequality we have $\mathbf{Pr}[\sum_i |\overline{J_i}| \geq 20k \cdot CC(\Pi)] \leq 1/10$. Therefore

$$\mathop{\mathbf{Pr}}_{R \sim \mathcal{R}^{\text{leaf}}} \left[\sum_i |\overline{J_i}|^2 \geq 400 \ (k \cdot CC(\Pi))^2 \right] \leq \mathbf{Pr} \left[\left(\sum_i |\overline{J_i}| \right)^2 \geq 400 \cdot (k \cdot CC(\Pi))^2 \right] \leq 1/10. \qquad \square$$

We will need the following claim about dense independent random variables.

Proposition 6.7. *Suppose that $X_1, \ldots, X_\ell$ are independent $(1-\delta)$-dense random variables where each X_i is supported on $[n^{1/k}]^{J_i}$ and $\delta = 1/\log n$. Then for any fixed $s \neq t \in [m]$ such that $\{s,t\} \cap J_i \neq \varnothing$ for all $i \in [\ell]$, the probability that $X_i(s) = X_i(t)$ for all $i \in [\ell]$ is at most $2n^{-\ell/k}$.*

Proof. Suppose without loss of generality that $t \in J_i$. Let $\alpha = X_i(s)$. Using the definition of the $(1-\delta)$-density X_i on $[n^{1/k}]^{J_i}$ with the set $I = \{t\}$ we have

$$\log \frac{1}{\mathbf{Pr}[X_i(t) = \alpha]} \geq (1-\delta)\log(n^{1/k}) = (1 - \frac{1}{\log n})\frac{\log n}{k} = \frac{\log n - 1}{k}$$

Therefore $\mathbf{Pr}[X_i(t) = \alpha] \leq (2/n)^{1/k}$. The final bound follows by independence and the fact that $\ell \leq k$. $\square$

We now focus on errors in protocols Π specifically for $\mathsf{Coll}^k_{m,n^{1/k}}$.

Definition 6.8. *Let R be a subrectangle in the support of $\mathcal{R}^{\text{leaf}}$. Call R* bad *if either*

- *there exists a pair $s \neq t \in [m]$ such that $s, t \in \bigcap_i \overline{J_i}$ and $\tau_i(s) = \tau_i(t)$ for all $i \in [k]$, or*
- $\sum_i |\overline{J_i}|^2 \geq 400\ (k \cdot CC(\Pi))^2$.

Otherwise call R good.

Note that, since we are focussed on proving a lower bound, our definition labels subrectangles as bad if they correspond to confirmed successes of Π.

Lemma 6.9. *If $CC(\Pi) \leq 0.01 n^{1/2 - 1/2k}/k$, then $\mathbf{Pr}_{R \sim \mathcal{R}^{\text{leaf}}}[R \text{ is bad}] < 1/6$.*

Proof. First, we observe that the probability in question, $\mathbf{Pr}_{R \sim \mathcal{R}^{\text{leaf}}}[R \text{ is bad}]$, is equivalent to the probability over a uniformly random input x that x ends up in a bad subrectangle, since our sampling is always proportional to the size of the subrectangles.

For simplicity we assume without loss of generality that Π is a full binary tree. Define $\mathcal{R}^\ell$ to be the support of subrectangles at the ℓ-th layer of the protocol tree in Algorithm 3, and let $R^\ell(x)$ (which has corresponding $J_1, \ldots, J_k$ and $\tau_1, \ldots, \tau_k$) be the subrectangle containing x in $\mathcal{R}^\ell$.

Define $G^\ell \subseteq \mathcal{R}^\ell$ to be the set of $R \in \mathcal{R}^\ell$ such that $\sum_i |\overline{J_i}|^2 < 400\ (k \cdot CC(\Pi))^2 \leq 0.04 n^{1-1/k}$ and let

$$B^\ell(x) = \mathbb{1}\{R^\ell(x) \notin G^\ell \text{ or } \exists\ s \neq t \in [m] \text{ and } j \in [k] \text{ s.t. } \tau_j(s) = \tau_j(t) \text{ and } x_{j'}(s) = x_{j'}(t) \text{ for all } j' \neq j\}.$$

We shall prove by induction on the layers of the sampling process given by Algorithm 3 that

$$\sum_x B^\ell(x) \leq \sum_{R \in G^\ell} \frac{2}{n^{1-1/k}} \sum_{i \in [k]} \binom{|\overline{J_i}|}{2} \cdot |R| + 1.04 \sum_{R \notin G^\ell} |R|. \tag{2}$$

The base case is clearly true, since at the root we have $\sum_x B^0(x) = 0$. Assume that (2) is true up to level ℓ. For every $R \notin G^\ell$, none of its subrectangles will be in $G^{\ell+1}$ since we only ever increase the size of $|\overline{J_i}|$ in our sampling process.

For the $R \in G^\ell$, suppose that player i speaks on R at level ℓ. We give convenient names to some of the objects produced during the sampling process. Algorithm 3 first partitions R into R^b for $b \in \{0,1\}$, and then further partitions R^b using the density restoring partition into $R^b_1, \ldots, R^b_r$, where each R^b_t has corresponding I^b_t new fixed coordinates for player i. In this way, for every $R \in \mathcal{R}^\ell$ we have that $|R| = \sum_{b \in \{0,1\}} \sum_{t=1}^r |R^b_t|$, since these form a partition.

Its possible that some of these subrectangles are no longer in $G^{\ell+1}$. We bound the size of these trivially. For those in $G^{\ell+1}$, using Proposition 6.7 we can bound the probability that the fixed variables for player i up level $\ell + 1$ include a collision on $s \neq t \in [m]$ that extends to a collision for all players. Fixing the coordinates in I^b_t creates at most $\binom{|\overline{J_i} \cup I^b_t|}{2} - \binom{|\overline{J_i}|}{2}$ fixed colliding pairs for player i that have not previously been fixed collisions for other players. By Proposition 6.7 and a union bound over these (s,t) pairs, the fraction of inputs in R^b_t that have collisions involving these pairs that have not previously been accounted for is at most

$$\frac{2}{n^{1-1/k}} \left(\binom{|\overline{J_i} \cup I^b_t|}{2} - \binom{|\overline{J_i}|}{2} \right).$$

Thus, we have that

$$\begin{aligned}
\sum_x B^{\ell+1}(x) &\leq \sum_x B^\ell(x) + \sum_{R \in G^\ell} \sum_{b \in \{0,1\}} \sum_{t : R^b_t \in G^{\ell+1}} \frac{2}{n^{1-1/k}} \left(\binom{|\overline{J_i} \cup I^b_t|}{2} - \binom{|\overline{J_i}|}{2} \right) |R^b_t| + \sum_{t : R^b_t \notin G^{\ell+1}} |R^b_t| \\
&\leq \sum_{R \in G^\ell} \frac{2}{n^{1-1/k}} \sum_{i \in [k]} \binom{|\overline{J_i(R)}|}{2} \cdot \sum_b \sum_t |R^b_t| + 1.04 \sum_{R \notin G^\ell} |R|
\end{aligned}$$

$$
\begin{aligned}
&+ \sum_{R\in G^\ell} \sum_{b\in\{0,1\}} \sum_{t:R_t^b\in G^{\ell+1}} \frac{2}{n^{1-1/k}} \left(\binom{|\overline{J_i(R)} \cup I_t^b|}{2} - \binom{|\overline{J_i(R)}|}{2} \right) |R_t^b| + \sum_{t:R_t^b\notin G^{\ell+1}} |R_t^b| \\
&= \sum_{R\in G^\ell} \sum_b \sum_{t:R_t^b\in G^{\ell+1}} |R_t^b| \frac{2}{n^{1-1/k}} \sum_{i\in[k]} \binom{|\overline{J_i(R_t^b)}|}{2} + 1.04 \sum_{R\notin G^\ell} \sum_b \sum_t |R_t^b| \\
&\quad + \sum_{R\in G^\ell} \sum_b \sum_{t:R_t^b\notin G^{\ell+1}} \left(1 + \frac{2}{n^{1-1/k}} \sum_{i\in[k]} \binom{|\overline{J_i(R)}|}{2} \right) |R_t^b| \\
&\leq \sum_{R\in G^{\ell+1}} \frac{2}{n^{1-1/k}} \sum_{i\in[k]} \binom{|\overline{J_i(R)}|}{2} |R| + 1.04 \sum_{R\notin G^{\ell+1}} |R|,
\end{aligned}
$$

as desired. The first inequality used a union bound and Proposition 6.7, as described above, and the second inequality used the inductive hypothesis. The third line recollected terms, while the last inequality used the fact that

$$
\frac{2}{n^{1-1/k}} \sum_{i\in[k]} \binom{|\overline{J_i(R)}|}{2} \leq 0.04,
$$

since $R \in G^\ell$.

Finally, we let $\ell = CC(\Pi)$. By Corollary 6.6, we can bound $1.04 \sum_{R\notin G^{CC(\Pi)}} |R| \leq 0.104 \sum_{R\in\mathcal{R}^{\text{leaf}}} |R| = 0.104 n^m$. We can also bound

$$
\sum_{R\in G^{CC(\Pi)}} \frac{2}{n^{1-1/k}} \sum_{i\in[k]} \binom{|\overline{J_i(R)}|}{2} |R| \leq 0.04 \sum_{R\in\mathcal{R}^{\text{leaf}}} |R| = 0.04 n^m.
$$

Combining, we get that the probability over uniformly random input of landing in a bad rectangle is at most $0.144 < 1/6$. □

Next, we have a simple lemma which says that the probability of producing a correct answer in any good leaf subrectangle is small.

Lemma 6.10. *Suppose that $k \leq \log n \ /4$. For any fixed $s \neq t \in [m]$ and good subrectangle R, the probability $\mathbf{Pr}_{x\in R}[x_i(s) = x_i(t)$ for all $i \in [k]] \leq 1/8$.*

Proof. We use Proposition 6.7 with $\ell = 1$. In a good rectangle, for any pair $s, t \in [m]$, there must be some J_i such that $\{s, t\} \not\subseteq \overline{J_i}$, in which case we can apply Proposition 6.7 with $\ell = 1$ to say that the probability that $x_i(s) = x_i(t) \leq 2n^{-1/k} \leq 2n^{-4/\log n} = 1/8$. □

We are now ready to prove Theorem 1.3.

Proof of Theorem 1.3. Suppose that the protocol Π communicated fewer than $0.01 n^{1/2-1/2k}/k$ bits. Then the probability of correctly outputting a collision is at most the probability of ending up in a bad subrectangle, plus the probability of outputting a correct collision given the input falls in a good subrectangle. We can thus bound the success probability of the protocol using Lemma 6.9 and Lemma 6.10 as at most $1/6 + 1/8 < 1/3$. □

7 Discussion

We end by discussing some related problems and directions for future study. In particular, we highlight three possible directions.

1. **Multiparty DAG-like communication lower bounds.** In general, it would be interesting to see if the $2^{\Omega(n^{1/4})}$ we give can be further improved, or if any nontrivial (size $2^{o(n)}$) cutting planes refutation

of BPHP_n^{n+1} exists. One possible way to improve the lower bound could be to study a multiparty ($k > 2$) version of DAG-like protocols. Concretely, could one can prove DAG-like communication lower bounds for the k-player analog $\mathsf{Search}^k_{\mathrm{BPHP}_n^m}$? We have shown that generalizing to k players helps in the tree-like case, and perhaps this holds true in the DAG-like setting as well.

2. **Stronger bounds for tree-like cutting-planes proof complexity of the weak bit pigeonhole principle** When $m = n + \Omega(n)$, we only obtain the weaker $2^{\Omega(\sqrt{n})}$ size lower bound for BPHP_n^m, rather than the $2^{n-o(1)}$ size lower bound we obtain for values of m closer to n. Can we extend the range of our strong lower bound to all values of $m > n$? To do so we must use a technique that does not use randomized communication complexity.

3. Finally, we highlight that for m close to n, the loss of 2^k in the denominator of Theorem 1.1 could potentially be improved when $m \leq n + n^{1/k}$. Indeed, we do not suspect that it should be present at all, and we conjecture that $\mathsf{Search}^k_{\mathrm{BPHP}_n^m}$ should remain hard for k all the way up to $\log n$. However, it seems unlikely that any reduction from k-party disjointness would be able to achieve this since an additional input bit per player seems essential in implementing the reduction.

References

[1] Albert Atserias, Moritz Müller, and Sergi Oliva. "Lower Bounds for DNF-Refutations of a Relativized Weak Pigeonhole Principle". In: *J. Symb. Log.* 80.2 (2015), pp. 450–476. DOI: 10.1017/JSL.2014.56. URL: https://doi.org/10.1017/jsl.2014.56.

[2] Paul Beame, Toniann Pitassi, and Nathan Segerlind. "Lower Bounds for Lovász–Schrijver Systems and Beyond Follow from Multiparty Communication Complexity". In: *SIAM J. Comput.* 37.3 (2007), pp. 845–869. DOI: 10.1137/060654645. URL: https://doi.org/10.1137/060654645.

[3] Paul Beame and Michael Whitmeyer. "Multiparty Communication Complexity of Collision Finding". In: *CoRR* abs/2411.07400 (2024). DOI: 10.48550/ARXIV.2411.07400. URL: https://doi.org/10.48550/arXiv.2411.07400.

[4] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. "Lower bounds for cutting planes proofs with small coefficients". In: *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June 1995, Las Vegas, Nevada, USA*. ACM, 1995, pp. 575–584. DOI: 10.1145/225058.225275. URL: https://doi.org/10.1145/225058.225275.

[5] Mark Braverman and Rotem Oshman. "The Communication Complexity of Number-In-Hand Set Disjointness with No Promise". In: *Electron. Colloquium Comput. Complex.* TR15-002 (2015). ECCC: TR15-002. URL: https://eccc.weizmann.ac.il/report/2015/002.

[6] Samuel R. Buss and György Turán. "Resolution Proofs of Generalized Pigeonhole Principles". In: *Theor. Comput. Sci.* 62.3 (1988), pp. 311–317. DOI: 10.1016/0304-3975(88)90072-2. URL: https://doi.org/10.1016/0304-3975(88)90072-2.

[7] Arkadev Chattopadhyay and Toniann Pitassi. "The story of set disjointness". In: *SIGACT News* 41.3 (2010), pp. 59–85. DOI: 10.1145/1855118.1855133. URL: https://doi.org/10.1145/1855118.1855133.

[8] Stefan S. Dantchev, Nicola Galesi, Abdul Ghani, and Barnaby Martin. "Proof Complexity and the Binary Encoding of Combinatorial Principles". In: *SIAM J. Comput.* 53.3 (2024), pp. 764–802. DOI: 10.1137/20M134784X. URL: https://doi.org/10.1137/20m134784x.

[9] Stephan Gocht, Jakob Nordström, and Amir Yehudayoff. "On Division Versus Saturation in Pseudo-Boolean Solving". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. ijcai.org, 2019, pp. 1711–1718. DOI: 10.24963/IJCAI.2019/237. URL: https://doi.org/10.24963/ijcai.2019/237.

[10] Mika Göös and Siddhartha Jain. "Communication Complexity of Collision". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2022, Urbana-Champaign, USA (Virtual Conference)*. Vol. 245. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 19:1–19:9. DOI: 10.4230/LIPICS.APPROX/RANDOM.2022.19. URL: https://doi.org/10.4230/LIPIcs.APPROX/RANDOM.2022.19.

[11] Mika Göös, Toniann Pitassi, and Thomas Watson. "Query-to-Communication Lifting for BPP". In: *SIAM J. Comput.* 49.4 (2020). DOI: 10.1137/17M115339X. URL: https://doi.org/10.1137/17M115339X.

[12] Dima Grigoriev, Edward A. Hirsch, and Dmitrii V. Pasechnik. "Complexity of Semi-algebraic Proofs". In: *STACS 2002, 19th Annual Symposium on Theoretical Aspects of Computer Science, Antibes - Juan les Pins, France, Proceedings*. Vol. 2285. Lecture Notes in Computer Science. Springer, 2002, pp. 419–430. DOI: 10.1007/3-540-45841-7_34. URL: https://doi.org/10.1007/3-540-45841-7%5C_34.

[13] Armin Haken. "The Intractability of Resolution". In: *Theor. Comput. Sci.* 39 (1985), pp. 297–308. DOI: 10.1016/0304-3975(85)90144-6. URL: https://doi.org/10.1016/0304-3975(85)90144-6.

[14] Armin Haken and Stephen A. Cook. "An Exponential Lower Bound for the Size of Monotone Real Circuits". In: *J. Comput. Syst. Sci.* 58.2 (1999), pp. 326–335. DOI: 10.1006/JCSS.1998.1617. URL: https://doi.org/10.1006/jcss.1998.1617.

[15] Johan Håstad. "On small-depth Frege proofs for PHP". In: *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*. IEEE, 2023, pp. 37–49. DOI: 10.1109/FOCS57990.2023.00010. URL: https://doi.org/10.1109/FOCS57990.2023.00010.

[16] Pavel Hrubes and Pavel Pudlák. "A note on monotone real circuits". In: *Electron. Colloquium Comput. Complex.* TR17-048 (2017). URL: https://eccc.weizmann.ac.il/report/2017/048.

[17] Pavel Hrubes and Pavel Pudlák. "Random formulas, monotone circuits, and interpolation". In: *Electron. Colloquium Comput. Complex.* TR17-042 (2017). ECCC: TR17-042. URL: https://eccc.weizmann.ac.il/report/2017/042.

[18] Russell Impagliazzo, Toniann Pitassi, and Alasdair Urquhart. "Upper and Lower Bounds for Tree-Like Cutting Planes Proofs". In: *Proceedings of the Ninth Annual Symposium on Logic in Computer Science (LICS '94), Paris, France, July 4-7, 1994*. IEEE Computer Society, 1994, pp. 220–228. DOI: 10.1109/LICS.1994.316069. URL: https://doi.org/10.1109/LICS.1994.316069.

[19] Dmitry Itsykson and Artur Riazanov. "Proof Complexity of Natural Formulas via Communication Arguments". In: *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*. Vol. 200. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, 3:1–3:34. DOI: 10.4230/LIPICS.CCC.2021.3. URL: https://doi.org/10.4230/LIPIcs.CCC.2021.3.

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

[21] Jan Krajícek, Pavel Pudlák, and Alan R. Woods. "An Exponential Lower Bound to the Size of Bounded Depth Frege Proofs of the Pigeonhole Principle". In: *Random Struct. Algorithms* 7.1 (1995), pp. 15–40. DOI: 10.1002/RSA.3240070103. URL: https://doi.org/10.1002/rsa.3240070103.

[22] Saburo Muroga. *Threshold logic and its applications*. Wiley, 1971. ISBN: 978-0-471-62530-8.

[23] Saburo Muroga, Iwao Toda, and Satoru Takasu. "Theory of majority decision elements". In: *Journal of the Franklin Institute* 271.5 (1961), pp. 376–418.

[24] Noam Nisan. "The communication complexity of threshold gates". In: *Combinatorics, Paul Erdos is Eighty* 1.301-315 (1993), p. 6.

[25] Toniann Pitassi, Paul Beame, and Russell Impagliazzo. "Exponential Lower Bounds for the Pigeonhole Principle". In: *Comput. Complex.* 3 (1993), pp. 97–140. DOI: 10.1007/BF01200117. URL: https://doi.org/10.1007/BF01200117.

[26] 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.

[27] Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020. DOI: 10.1017/9781108671644.

[28] Ran Raz and Avi Wigderson. "Monotone Circuits for Matching Require Linear Depth". In: *J. ACM* 39.3 (1992), pp. 736–744. DOI: 10.1145/146637.146684. URL: https://doi.org/10.1145/146637.146684.

[29] Hanif D. Sherali and Warren P. Adams. "A Hierarchy of Relaxations Between the Continuous and Convex Hull Representations for Zero-One Programming Problems". In: *SIAM J. Discret. Math.* 3.3 (1990), pp. 411–430. DOI: 10.1137/0403036. URL: https://doi.org/10.1137/0403036.

[30] Dmitry Sokolov. "Dag-Like Communication and Its Applications". In: *Computer Science - Theory and Applications - 12th International Computer Science Symposium in Russia, CSR 2017, Kazan, Russia, June 8-12, 2017, Proceedings*. Vol. 10304. Lecture Notes in Computer Science. Springer, 2017, pp. 294–307. DOI: 10.1007/978-3-319-58747-9_26. URL: https://doi.org/10.1007/978-3-319-58747-9%5C_26.

[31] 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 2024, Vancouver, BC, Canada, June 24-28, 2024*. ACM, 2024, pp. 2008–2015. DOI: 10.1145/3618260.3649636. URL: https://doi.org/10.1145/3618260.3649636.

[32] Emanuele Viola. "The communication complexity of addition". In: *Combinatorica* 35 (2015), pp. 703–747. DOI: 10.1007/S00493-014-3078-3.

[33] Shuo Wang, Guangxu Yang, and Jiapeng Zhang. "Communication Complexity of Set-Intersection Problems and Its Applications". In: *Electron. Colloquium Comput. Complex.* TR23-164 (2023). ECCC: TR23-164. URL: https://eccc.weizmann.ac.il/report/2023/164.

[34] 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 2024, Vancouver, BC, Canada, June 24-28, 2024*. ACM, 2024, pp. 630–639. DOI: 10.1145/3618260.3649607. URL: https://doi.org/10.1145/3618260.3649607.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il