Hardness of Constant-Round Communication Complexity

Shuichi Hirahara
National Institute of Informatics, Tokyo, Japan

Rahul Ilango
Massachusetts Institute of Technology, Cambridge, MA, USA

Bruno Loff
INESC-Tec and University of Porto, Portugal

Abstract

How difficult is it to compute the communication complexity of a two-argument total Boolean function $f : [N] \times [N] \to \{0, 1\}$, when it is given as an $N \times N$ binary matrix? In 2009, Kushilevitz and Weinreb showed that this problem is cryptographically hard, but it is still open whether it is NP-hard.

In this work, we show that it is NP-hard to approximate the size (number of leaves) of the smallest *constant-round* protocol for a two-argument total Boolean function $f : [N] \times [N] \to \{0, 1\}$, when it is given as an $N \times N$ binary matrix. Along the way to proving this, we show a new *deterministic* variant of the round elimination lemma, which may be of independent interest.

2012 ACM Subject Classification Theory of computation → Communication complexity; Theory of computation → Problems, reductions and completeness

Keywords and phrases NP-completeness, Communication Complexity, Round Elimination Lemma, Meta-Complexity

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.31

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2021/030/

Funding *Shuichi Hirahara*: This work was partly carried out during a visit supported by ACT-I, JST.
Rahul Ilango: During this work, this author was funded by an Akamai Presidential Fellowship and by NSF Grants CCF-1741615 and CCF-1909429.
Bruno Loff: This project was financed by the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project UIDB/50014/2020. This work was partly carried out during a research visit conducted with support from DIMACS in association with its Special Focus on Lower Bounds.

Acknowledgements The authors would like to thank Ryan Williams for his support, and for several discussions and suggestions, without which this paper would not have existed. The authors would also like to thank Igor Oliveira for helpful conversations about hardness of communication complexity.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 31; pp. 31:1–31:30
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Contents

1 Introduction 2

1.1 Motivation . . . 2

1.2 Our Results . . . 3

1.3 Meta-Complexity . . . 5

1.3.1 Previous work . . . 6

1.4 Outline of the Paper . . . 7

1.5 Sketch of Lemma 3 . . . 7

1.6 Concluding remarks and open problems . . . 9

2 Preliminaries 9

3 Warmup: deterministic 3-round protocols, large output alphabet 11

4 Hardness for deterministic 3-round protocols 14

5 From 3-rounds to multiple rounds using deterministic round elimination 16

6 Hardness for randomized 3-round protocols 19

7 From 3-rounds to multiple rounds using round elimination 24

1 Introduction

Suppose you are given a $N \times N$ Boolean matrix representing a (total) two-player communication problem. How difficult is it to determine the (deterministic) communication complexity of this matrix?

In 2009, Kushilevitz and Weinreb [39] studied this question and showed that, under a cryptographic assumption, no polynomial-time algorithm can compute the communication complexity of a given total two-player function. They left open the question of whether this problem is NP-hard.

Our main result is that the problem of determining the minimum number of leaves in an *d-round* communication protocol for a given (total, two player) function is NP-hard, for all integer constants $d \geq 3$.

1.1 Motivation

Determining the difficulty of computing communication complexity is an interesting, basic question in its own right. However, the aforementioned paper of Kushilevitz and Weinreb – which gave the first non-trivial results on this problem – was also motivated by the broader implications this question could have for communication complexity. This fits into an even broader motif that has become prominent in recent years: using "meta-questions" to investigate various aspects of complexity theory.

For example, Kushilevitz and Weinreb argue that understanding the intractability of computing communication complexity can help "explain the difficulty of analyzing the communication complexity of certain functions." Towards this end, their cryptographic hardness result exhibits a family of functions whose communication complexity we are unlikely to ever gain a complete understanding of (since determining their communication complexity is crytographically hard).

Kushilevitz and Weinreb also used the meta-complexity lens to shed light on one of the oldest questions in communication complexity: the log-rank conjecture of Lovasz and Saks [41]. If the log-rank conjecture is true, then it yields a simple polynomial-time approximation algorithm for computing communication complexity (simply output the logarithm of the rank of the input matrix). A natural question is whether one can get a better approximation algorithm. Kushilevitz and Weinreb introduced a plausible conjecture that would imply that the log-rank conjecture, if true, yields an optimal polynomial-time approximation. On the other hand, a strong enough hardness of approximation result could actually disprove the log-rank conjecture (conditioned on P $\neq$ NP). Thus, understanding the inapproximability of computing communication complexity seems closely related to resolving the log-rank conjecture.

Finally, Kushilevitz and Weinreb's paper also introduced a remarkable new technique, showing for the first time how one could devise a total two-player Boolean function whose communication complexity was strongly tied to the Boolean-formula complexity of another, related function. Prior to their work, connections had only been known between Boolean-formula complexity and the communication complexity of *search* problems [35] and were not known for *decision* problems.

Thus, it is plausible that proving NP-hardness results for computing communication complexity could reveal further insights in communication complexity and lead to the development of useful new techniques. Indeed, our constant-round NP-hardness result led us to prove an interesting new direct-sum/round-elimination result in deterministic communication complexity, which we state in the following section.

1.2 Our Results

In order to state our results formally, we fix some notation. If $f : [a] \times [b] \to \{0,1\}$ is a two-player Boolean-valued function, then

- $\mathsf{C}^{\mathrm{A}}_d(f)$ denotes its d-round deterministic communication complexity, namely, the smallest number of bits communicated in a d-round protocol that computes f, where Alice speaks in the first round,
- $\mathsf{L}^{\mathrm{A}}_d(f)$ denotes the minimum number of leaves in a d-round protocol that computes f where Alice speaks first,
- $\mathsf{C}^{\mathrm{B}}_d(f)$ and $\mathsf{L}^{\mathrm{B}}_d(f)$ denote the analogous notions where Bob speaks first, and
- $\mathsf{C}^{\mathrm{A}}_{d,\varepsilon}(f)$, $\mathsf{C}^{\mathrm{B}}_{d,\varepsilon}(f)$, $\mathsf{L}^{\mathrm{A}}_{d,\varepsilon}(f)$ and $\mathsf{L}^{\mathrm{B}}_{d,\varepsilon}(f)$ denote the analogous notions but where the protocol is probabilistic, and is allowed to err with probability $\leq \varepsilon$.

Our first result shows that computing 3-round deterministic communication complexity is NP-hard. We construct a reduction from the chromatic number problem to the problem of computing 3-round deterministic communication complexity. Our reduction attains the following hardness:

▶ **Theorem 1** (Informal version of Theorem 18)**.** *It is* NP*-hard to approximate* $\mathsf{L}^{\mathrm{A}}_3(f)$ *to within a factor of* $N^{1/8}$*, and* $\mathsf{C}^{\mathrm{A}}_3(f)$ *to within an additive term of* $\frac{1}{8}\log N$*,*[1] *when given a function* $f : [N] \times [N] \to \{0,1\}$*.*

We then work to prove NP hardness for all constants $d \geq 3$ by induction on d, using Theorem 1 as a base case. Thus, in our inductive step, our goal is to show that computing d-round communication complexity reduces to computing $(d+1)$-round communication complexity.

[1] Since $\mathsf{C}^{\mathrm{A}}_3(f) \leq \log N + 1$, this implies it is hard to approximate $\mathsf{C}^{\mathrm{A}}_3(f)$ to within a multiplicative factor of $1 + \frac{1}{8}$.

A natural approach would be to use the round elimination lemma [46, 62]. This lemma says that given a two-player function f, one can create a new function F such that the $(d+1)$-round communication complexity of F is closely related to the d-round communication complexity of f.

There are a few difficulties in using round elimination. For one, going from f to F in round elimination requires a dramatic blow up of the input size of the function. As a result, any reduction based on typical round elimination seems to require a superpolynomial running time.

A more significant issue is that round elimination only works for probabilistic protocols, not deterministic protocols. So, in order to use round elimination, we would actually need a much stronger version of Theorem 1 for our base case: that it is hard to distinguish protocols that have small three-round *deterministic* communication complexity from protocols that require large three-round *randomized* communication complexity. As it turns out, we can "almost" prove such a result (see Section 6):

▶ **Theorem 2.** *There exist positive constants γ and δ such that the following holds. There exists a deterministic quasipolynomial-time algorithm that, on input $x \in \{0,1\}^*$, outputs a communication matrix $M \in \{0,1\}^{N\times N}$ and a number $k \in \mathbb{N}$, with $k \le N = |x|^{O(1)}$, such that*

1. *if x is a YES instance of SAT, then $\mathsf{L}_3^{\mathrm{B}}(M) \le O(k)$ and $\mathsf{C}_3^{\mathrm{B}}(M) \le \log k + O(1)$, and*
2. *if x is a NO instance of SAT, then $\mathsf{L}_{3,N^{-\delta}}^{\mathrm{B}}(M) \ge \Omega(N^{\gamma}\cdot k)$ and $\mathsf{C}_{3,N^{-\delta}}^{\mathrm{B}}(M) \ge \log k + \gamma \cdot \log N - O(1)$.*

Unfortunately, the hardness parameters we obtain are not enough to make the round-elimination approach work. If in the above theorem we could have chosen $\gamma \ge c\cdot\delta$ for an arbitrarily large constant c, then we would be able to use round elimination to show that $\mathsf{C}_d^{\mathrm{A}}(f)$ is NP-hard for any constant number of rounds d, under subexponential-time reductions (this is proven in Section 7). If we could make the error parameter constant instead of $N^{-\delta}$, then we would be able to show that $\mathsf{C}_d^{\mathrm{A}}(f)$ is NP-hard under quasipolynomial-time reductions. We leave proving a version of Theorem 2 with these stronger parameters as an open problem.

In light of these difficulties, a natural question is whether there exists an alternative to round elimination that works with deterministic protocols. Ideally, this alternative method would also avoid introducing a subexponential blowup. Towards this end, we prove a new result in deterministic communication complexity that gives us a tight relation between the minimum number of leaves in a d-round protocol for f, and the minimum number of leaves in a $(d+1)$-round protocol for a related function F. This function F is a kind of "direct sum" of f with the "XOR-equality" function. It should be remarked that the direct-sum property is known to *fail* for general deterministic protocols [56], so we cannot, for example, replace XOR-equality with another arbitrary function with the same communication complexity. The formal statement of our result is as follows:

▶ **Lemma 3.** *Let $d \ge 3$. Given an arbitrary two-player total Boolean function $f : [a]\times[b] \to \{0,1\}$, define the function $F : ([k]\times[a]) \times ([k]\times[b]\times\{0,1\}\times\{0,1\}) \to \{0,1\}$ given by*

$$F(x_0,x_1;y_0,y_1,z,i) = \begin{cases} \mathsf{XorEq}_k(x_0;y_0,z) & , \text{ if } i=0 \\ f(x_1;y_1) & , \text{ if } i=1, \end{cases}$$

where, in turn, $\mathsf{XorEq}_k : [k]\times([k]\times\{0,1\}) \to \{0,1\}$ is given by

$$\mathsf{XorEq}_k(x;y,z) = \begin{cases} z & \text{if } x \ne y \\ 1-z & \text{if } x = y. \end{cases}$$

Then

$$\min\{4k, 2k-2+\mathsf{L}_d^{\mathrm{B}}(f)\} \le \mathsf{L}_{d+1}^{\mathrm{A}}(F) \le 2k + \mathsf{L}_d^{\mathrm{B}}(f).$$

The last two inequalities can be seen as saying that $\mathsf{L}^{\mathrm{A}}_{d+1}(F) - 2k$ is a good approximation of $\mathsf{L}^{\mathrm{B}}_{d}(f)$, for $k \geq \mathsf{L}^{\mathrm{B}}_{d}(f)$. Thus, it is natural to view Lemma 3 not just as a direct-sum-type result, but also as a kind of round elimination lemma, since it relates the $(d+1)$-round communication complexity of F with the d-round communication complexity of f.

Lemma 3 has a few significant differences from the classical round elimination lemma. First, while the classical lemma only applies to probabilistic protocols, Lemma 3 works in the deterministic case.

Second, Lemma 3 is more efficient in the number of inputs of F relative to f. Using the classical round elimination lemma would require at least a quasipolynomial blowup in going from f to F, but a quadratic blowup suffices for Lemma 3. This allows us to build a polynomial-time NP-hardness reduction instead of the superpolynomial-time reduction that would follow from the randomized round-elimination approach.

Third, our proof of Lemma 3 looks very different than the classical proof of the round elimination lemma, which is almost entirely information-theoretic.[2] Our proof is instead more combinatorial and builds on a fooling set argument. We sketch the proof of Lemma 3 in Section 1.5.

Using Theorem 1 and Lemma 3, we obtain our main theorem:

▶ **Theorem 4.** *For any $d \geq 3$ there exists a constant $\Delta_d > 0$ such that the following holds. If there exists a polynomial-time algorithm which, when given a total two-player Boolean-valued function $f : [N] \times [N] \to \{0,1\}$ represented as a Boolean matrix of dimensions $N \times N$, approximates $\mathsf{L}^{\mathrm{A}}_{d}(f)$ within a factor of $1 + \Delta_d$, then* $\mathsf{P} = \mathsf{NP}$.

1.3 Meta-Complexity

Our work fits into a now well-established theme in computational complexity theory of studying "meta-complexity questions." Historically, this kind of question was first studied by Soviet cyberneticians beginning in the 1950s, who were particularly interested in the problem of circuit minimization: the ("meta-complexity") task of computing the smallest circuit for a prescribed Boolean function [63]. At the time, this was considered to be among the least likely computational task to have better-than-brute-force algorithms. Reportedly, Levin delayed the publication of his work on NP-completeness, because he was hoping to show the NP-hardness of this problem [12].

Since then, meta-complexity questions have become so pervasive, that there are few unsolved problems in computational complexity which are not touched by meta-complexity results. For example:

- The relativization barrier [14], and related algebrization barrier [1], imply that a number of proof techniques will be insufficient to settle most uniform complexity-class separation questions.
- The natural proofs barrier [60, 50] also immediately excludes us from considering many properties which might, at a first glance, plausibly imply hardness of a given Boolean function for circuit classes above TC_0.
- It is known that the complexity measures we are interested in understanding, such as the number of leaves in Boolean formulae, are inherently non-convex [34] and non-submodular [59], and thus cannot, for example, be approximated by convex programming or by certain rank-based measures.

[2] Indeed, using information theoretic techniques, like the chain rule and Pinsker's inequality, seems to require a superpolynomial blowup.

- Lower-bounds that mildly improve classic lower-bounds (e.g. super-linear lower-bounds against NC_1 [11], lower-bounds against one-pass streaming algorithms [44]) or lower-bounds for certain problems (e.g. k-vertex cover [54, 55]) against complexity classes for which we already have lower-bounds, could be "magnified" to solve longstanding open problems.
- Just recently, the existence of one-way functions is now known to be equivalent to the average-case hardness of computing polytime-bounded Kolmogorov complexity [40].

In this paper, we contribute to one of the lines of research inscribed in this theme. Generically, fixing a complexity measure $\mathcal{C}$, we can define a "meta"-problem $\mathsf{MP}_{\mathcal{C}}$ where a task T is the input to the problem $\mathsf{MP}_{\mathcal{C}}$, and the problem $\mathsf{MP}_{\mathcal{C}}$ is to compute $\mathcal{C}(T)$, namely, the $\mathcal{C}$-complexity of T. The "meta"-question is then: what is the computational complexity of $\mathsf{MP}_{\mathcal{C}}$?

1.3.1 Previous work

This meta-question has been studied in many previous works. Most of these works deal with the case where T is the truth table of a Boolean function and the complexity measure $\mathcal{C} = \mathsf{SIZE}$ is the size of the smallest circuit computing T; in this case $\mathsf{MP}_{\mathsf{SIZE}}$ is denoted MCSP, which stands for "Minimum Circuit-Size Problem." The question originally posed by Levin is whether MCSP is NP-complete, and this is the main unresolved question in this area.

We seem far from settling this question, but MCSP is known to be hard for various other classes [54, 22, 54, 5, 6]. It is also known that MCSP is *not* NP-hard under various weak reductions [48, 33, 48, 27, 26, 8, 10]. MCSP has many natural connections to other areas, such as cryptography [57, 61], natural proofs [61], hardness magnification [53, 45], learning [18], and proof complexity [37, 47]). A few variants of MCSP are known to be NP-hard, including some relativized versions of MCSP [9, 26, 31], a conditional version of MCSP [28], and MCSP for multi-output functions [30]. For more information on recent research, see Allender's recent survey [4] and the references therein.

Thus far, relatively few works have focused on proving the NP-hardness of computing other complexity measures. The NP-hardness of computing the size of the smallest DNF for a given function (given as a truth table) was first established by Masek, already in 1978 [43]. A series of subsequent works later improved this result to give near-optimal hardness-of-approximation [19, 64, 21, 7, 36]. More recently, it was shown to be NP-hard to compute the size of the smallest DNF-of-XORs [25]. Other works have established NP-hardness of the task of finding an optimal algorithm, such as finding the smallest decision tree for a given partial function [23], finding the weights of a neural network (with a fixed topology) that computes a given function [32, 16], or finding the smallest straight-line program for computing a given linear form [17].

A special reference should be made to the previous work by one of the authors [29], where it was shown that it is NP-hard to approximate the size of the smallest AC^0-formula of a total function given as a truth-table, with an approximation factor of $1 \pm \Delta_d$, where Δ_d depends on the depth d. Our paper is inspired by this earlier paper, and our work can be seen as proving an analogous result for constant-round protocols for total Boolean functions, to the result proven in [29] for constant-round protocols for Karchmer–Wigderson games. The high-level idea of the proof is similar, as well: we first prove hardness for constant depth, and then show how to reduce the depth-d problem to the depth-$(d+1)$ problem. However, the required techniques are completely different, both for establishing the base case and the inductive step. The depth 2 problem is already hard in the case of minimizing Boolean formulas, but it can be solved in polynomial time in the case of communication complexity. Hence, our first hard case is the depth 3 case. Our proof of the inductive depth-d to depth-$(d+1)$ reduction is also very different, and we further discuss the differences in Section 1.5.

1.4 Outline of the Paper

Our proofs take the approach of starting with simple cases and building up to more complicated cases. We begin in Section 3 by showing the NP-hardness of approximating the 3-round communication complexity of total, *multi-output* functions, i.e., total two-player functions $f : [a] \times [b] \to [\ell]$. This problem has a nice combinatorial interpretation (see Proposition 11), and it turns out to be NP-hard to approximate by a simple reduction from graph coloring. The proof is an interesting combination of an NP-hardness reduction with a communication-complexity lower-bound argument.

We then prove in Section 4 that 3-round communication complexity is also hard to approximate for *Boolean* functions $f : [a] \times [b] \to \{0, 1\}$. The reduction is inspired by the multi-output case, but it requires us to devise a particular kind of gadget. The existence of such a gadget can be proven using the probabilistic method, but this only yields a randomized reduction. We show the probabilistic construction can instead be derandomized using efficient constructions of universal sets [51], and this results in a deterministic reduction. So this result combines an NP-hardness reduction, a communication complexity lower bound, and a derandomization result.

After proving hardness for 3 round protocols, it is natural to use the classical round-elimination technique to prove the result for any number of rounds. However, in order to use round-elimination, we need hardness of approximation for 3 round *randomized* communication complexity. We do show, in Section 6, that it is hard to approximate the 3-round *randomized* communication complexity in a low error regime, but the parameters we obtain are just shy of what would be necessary to show hardness of approximation for d-round communication complexity, for any d. We still conjecture that better parameters can be obtained, and we show in Section 7 that our conjecture would imply that d-round communication complexity is NP-hard to approximate, by a reduction from the $(d-1)$-round case. Improving the parameters in our lower bound is left as an open problem.

In Section 5, we finish the proof of our main theorem. There we show our deterministic round-elimination lemma (Lemma 3), and use it to show that the smallest *number of leaves* in a constant-round communication protocol is NP-hard to approximate.

1.5 Sketch of Lemma 3

In this subsection we sketch the proof of our deterministic round-elimination lemma, Lemma 3. Let us restate it here:

▶ **Lemma 3.** *Let $d \geq 3$. Given an arbitrary two-player total Boolean function $f : [a] \times [b] \to \{0, 1\}$, define the function $F : ([k] \times [a]) \times ([k] \times [b] \times \{0, 1\} \times \{0, 1\}) \to \{0, 1\}$ given by*

$$F(x_0, x_1; y_0, y_1, z, i) = \begin{cases} \mathsf{XorEq}_k(x_0; y_0, z) & , \text{ if } i = 0 \\ f(x_1; y_1) & , \text{ if } i = 1, \end{cases}$$

where, in turn, $\mathsf{XorEq}_k : [k] \times ([k] \times \{0, 1\}) \to \{0, 1\}$ is given by

$$\mathsf{XorEq}_k(x; y, z) = \begin{cases} z & \text{if } x \neq y \\ 1 - z & \text{if } x = y. \end{cases}$$

Then

$$\min\{4k, 2k - 2 + \mathsf{L}^{\mathrm{B}}_d(f)\} \leq \mathsf{L}^{\mathrm{A}}_{d+1}(F) \leq 2k + \mathsf{L}^{\mathrm{B}}_d(f).$$

The upper bound is the easy direction (it follows from XorEq_k having a $2k$ leaf 2-round Alice-first protocol), so here we focus on how to prove the lower bound.

High Level Ideas. At a high level, our lower bound proof works by showing that any protocol for computing F must do one of two things:

- compute XorEq_k "twice," or
- compute XorEq_k "once" and (mostly) separately compute f in d rounds.

These two scenarios correspond to the $4k$ and $2k - 2 + \mathsf{L}_d^{\mathsf{B}}(f)$ parts of the lower bound respectively.

It is worth noting that an approach similar to this was used in [29] to prove a lower bound that related the depth-d and depth-$(d+1)$ *formula complexity* of two functions. Indeed, our proof was partly inspired by the proof in [29]. We note, however, that the proof in [29] differs significantly in how it implements this high level approach. We highlight a few of these differences:

- The proof of the lower bound in [29] is based on the probabilistic method (in particular, showing that some random subformulas have nice properties). Our lower bound does not involve the probabilistic method and is instead based on fooling sets.
- [29] relies on a complicated formalization of computing g "twice" that involves computing a large one-sided approximation of g with a non-deterministic formula. On the other hand, our formalization of computing XorEq_k "twice" is to contain twice as many leaves as it would take to compute XorEq_k exactly in a three-round protocol.
- [29] uses a random function g instead of the XorEq_k function.
- Our lemma is tight up to an additive 2 term, while the lower bound in [29] is only known to be tight up to a multiplicative $(1 \pm o(1))$ factor.

Another important aspect of our proof is how we make use of the XorEq_k function. The key property used about the XorEq_k function is that it has a tight fooling set lower bound: i.e., a fooling set lower bound that shows it requires exactly $2k$ leaves to compute. (The equality function has a tight fooling set lower bound for its 1-leaves, but not for its 0-leaves. XorEq_k is a modification of the equality function that "symmetrizes" the function enough that we get a tight fooling set for both the 1 and 0 leaves.) This tight fooling set severely limits the structure of monochromatic combinatorial rectangles in F, which we use in both cases of our proof.

Proof Sketch. Suppose that π is a $(d+1)$-round Alice-first protocol for F. We split into cases depending on how Alice partitions her inputs in the first round of the protocol.

Recall that Alice gets an input $(x_0, x_1) \in [k] \times [a]$. Let $\mathcal{P} = \{P_1, \ldots, P_{|\mathcal{P}|}\}$ be Alice's partition of her inputs $[k] \times [a]$ in the first round of the protocol. We say that $\mathcal{P}$ is *good* if there exists a $x_0^\star \in [k]$ such that $\{x_0\} \times [a] \subseteq P_q$ for some q. (The reason behind choosing this to be our definition of good is that it implies that one can obtain a round d protocol for solving f by considering the subprotocol of π obtained by restricting $x_0 = x_0^\star$).

If $\mathcal{P}$ is not good, then for each $x_0 \in [k]$ there are distinct $x_1', x_1'' \in [a]$ such that (x_0, x_1') and (x_0, x_1'') are contained in different parts in Alice's partition. Consequently, any leaf in π that contains an input where Alice's input is (x_0, x_1') must be distinct from any leaf in π that contains an input where Alice's input is (x_0, x_1''). We combine this "distinct leaves" property with the fooling set for XorEq_k in order to show that the protocol must spend twice as many leaves as is optimal for computing XorEq_k. Intuitively, this is because when $i = 0$, F computes $\mathsf{XorEq}_k(x_0; y_0, z)$, which doesn't depend on the value of x_1. Thus, every element $(x_0; y_0, z)$ of a fooling set for XorEq_k can be used to produce two distinct leaves in π: one leaf for when Alice gets the input (x_0, x_1') and one leaf for when Alice gets the input (x_0, x_1'').

On the other hand, suppose $\mathcal{P}$ is good. Then $\{x_0^\star\} \times [a] \subseteq P_q$ for some q. As a result, the d round Bob-first subprotocol of π, given when Alice is restricted to an input in P_q, computes f when we set $x_0 = x_0^\star$ and $i = 1$. This implies that π must have at least $\mathsf{L}_d^{\mathrm{B}}(f)$ leaves.

In fact, the goodness of $\mathcal{P}$ implies a stronger statement: that π contains at least $\mathsf{L}_d^{\mathrm{B}}(f)$ many leaves which contain an input where $x_0 = x_0^\star$. This is crucial because only two elements of the fooling set for XorEq_k satisfy $x_0 = x_0^\star$. Consequently, one can show that, in order to compute XorEq_k when $i = 0$, π must have $2k-2$ leaves that do not contain any input where $x_0 = x_0^\star$. Putting the two bounds together, we get a $2k - 2 + \mathsf{L}_d^{\mathrm{B}}(f)$ lower bound on the number of leaves in π.

1.6 Concluding remarks and open problems

In this work we make a significant step towards showing that computing communication complexity is NP-hard, by proving it is hard to compute the smallest size of a constant-round protocol for a given function.

There are a few natural open questions suggested by our paper. The biggest question is whether the unbounded-round case is also NP-hard. But even in the constant-round setting, we would like to prove better hardness of approximation for protocol size, and we would like to prove unconditionally that communication complexity is NP-hard. Proving Conjecture 33 would give us both of these results for subexponential-time reductions. Can we get such hardness using polynomial-time reductions, as well? On the other hand, one can also ask: do there exist non-trivial polynomial-time approximation algorithms for computing constant-round communication complexity? It is worth noting that the log-rank conjecture gives a candidate approximation algorithm for computing communication complexity with no bound on the number of rounds.

A crucial ingredient in our hardness result is a deterministic version of the round elimination lemma, which is proved using entirely different techniques than the original version. Does this deterministic version have other applications? Can the new ideas in our proof be used to prove other interesting statements?

2 Preliminaries

For a positive integer n, we let $[n]$ denote the set $\{1, \dots, n\}$.

Binomial Coefficients and Projections. The binomial coefficient $\binom{n}{k}$ equals the number of distinct subsets of $[n]$ with exactly k elements. Similarly, $\binom{n}{\leq k}$ denotes the number of distinct subsets of $[n]$ that have at most k elements. Finally, $\binom{[n]}{k}$ denotes the set of all subsets of $[n]$ with exactly k elements.

If $x \in \{0,1\}^n$ and $S = \{i_1, \dots, i_k\} \subseteq \binom{[n]}{k}$, then $x_S \in \{0,1\}^k$ denotes the *projection of x to S*, given by $x_S = x_{i_1} \dots x_{i_k}$.

Entropy, Mutual Information, Pinsker's inequality. We describe the notation we will use for various information-theoretic quantities, and some basic facts about them. We will not define the notions here, or prove the basic facts. See [58] for a reference that uses these notions in the context of communication complexity. Given a random variable $\mathbf{x} \in X$, we denote its entropy by $\mathsf{H}(\mathbf{x})$. Given random variables $\mathbf{x}, \mathbf{y}, \mathbf{z}$, we will denote the mutual information between $\mathbf{x}$ and $\mathbf{y}$, given $\mathbf{z}$, by $\mathsf{I}(\mathbf{x} : \mathbf{y} \mid \mathbf{z})$. It always holds that $\mathsf{I}(\mathbf{x} : \mathbf{y}) \leq \mathsf{H}(\mathbf{y})$. If we have random variables $\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{y}$, we then have the *chain rule*:

$$\mathsf{I}(\mathbf{x}_1, \dots, \mathbf{x}_n : \mathbf{y}) = \sum_{i=1}^{n} \mathsf{I}(\mathbf{x}_i : \mathbf{y} \mid \mathbf{x}_{<i}),$$

where $\mathbf{x}_{<i} = \mathbf{x}_1, \ldots, \mathbf{x}_{i-1}$. If two random-variables $\mathbf{x}$ and $\mathbf{y}$ have $\mathsf{I}(\mathbf{x} : \mathbf{y}) \leq 2\varepsilon^2$, then Pinsker's inequality implies (by concavity) that if we compute the average, over the choice y for $\mathbf{y}$, statistical distance between the distribution of $\mathbf{x}$, and the distribution of $\mathbf{x}$ conditioned on $\mathbf{y} = y$, then this average is less than ε.

Communication complexity. We assume basic familiarity with communication complexity [38]. We write the definitions here for clarity.

▶ **Definition 5** (Protocol). *Let $\mathcal{A}, \mathcal{B}, \mathcal{Z}$ be finite sets. A* deterministic protocol *π over $\mathcal{A} \times \mathcal{B} \times \mathcal{Z}$ is a rooted tree:*

- *Each node v is associated with a rectangle $\pi^{-1}(v) = A \times B$, with $A \subseteq \mathcal{X}$ and $B \subseteq \mathcal{Y}$.*
- *Each non-leaf node v, associated with a rectangle $\pi^{-1}(v) = A \times B$, is labeled by either (a) a partition $A = \dot{\bigcup}_{c \in \mathcal{P}_v} A_c$ of A, in which case we say it is* Alice's node *or (b) a partition $B = \dot{\bigcup}_{c \in \mathcal{P}_v} B_c$ of B, in which case we say it is* Bob's node.
- *Each leaf node is labeled by an element of the output domain $\mathcal{Z}$*
- *The rectangle associated with the root is the input domain $\mathcal{A} \times \mathcal{B}$.*
- *If a non-leaf node v of Alice is associated with rectangle $\pi^{-1}(v) = A \times B$ and (a) labeled by a partition $A = \dot{\bigcup}_{c \in \mathcal{P}_v} A_c$ of A, then for each $c \in \mathcal{P}_v$ there will be one child v_c of v, which will be associated with the rectangle $\pi^{-1}(v_c) = A_c \times B$; similarly for Bob's nodes.*

We let the leaf complexity of π, *written* $\mathsf{L}(\pi)$, *be the number of leaf nodes of π. We let the* round complexity of π, *written* $\mathsf{R}(\pi)$, *be the height of π, i.e., the maximum number of edges in any root-to-leaf path of π.*

A root-to-leaf path $v_1 \to \cdots \to v_{k+1}$ is said to have communication length $\sum_{i=1}^{k} \lceil \log |P_{v_i}| \rceil$ *(where $|P_{v_i}|$, as defined above, is the number of parts in the partition of A or B associated with node v_i). We then let the* communication complexity of π, *written* $\mathsf{C}(\pi)$, *be the maximum communication length of any root-to-leaf path of π.*

Given $(a, b) \in \mathcal{A} \times \mathcal{B}$, we let $\pi(a, b)$ denote the (unique) leaf v of π having $(a, b) \in \pi^{-1}(v)$. For $z \in \mathcal{Z}$, we may write $\pi(a, b) = z$ to mean that the leaf $\pi(a, b)$ is labeled by z.

A randomized protocol *over $\mathcal{A} \times \mathcal{B} \times \mathcal{Z}$ is a distribution over deterministic protocols over $\mathcal{A} \times \mathcal{B} \times \mathcal{Z}$. We will use a boldface Greek letter, such as $\boldsymbol{\pi}$, to denote a protocol sampled from this distribution. We then let $\mathsf{L}(\boldsymbol{\pi})$ be the maximum $\mathsf{L}(\pi)$ over all π in the support of $\boldsymbol{\pi}$, and likewise for $\mathsf{R}(\boldsymbol{\pi})$ and $\mathsf{C}(\boldsymbol{\pi})$.*

▶ **Definition 6.** *A function $f : \mathcal{A} \times \mathcal{B} \to \mathcal{Z}$ is said to be* computed *by a deterministic protocol π over $\mathcal{A} \times \mathcal{B} \times \mathcal{Z}$ if we have $f(a, b) = \pi(a, b)$ for every $(a, b) \in \mathcal{A} \times \mathcal{B}$. Furthermore, if $\varepsilon \in [0, 1]$, then f is said to be* computed with error ε *by a randomized protocol $\boldsymbol{\pi}$ if, for every $(a, b) \in \mathcal{A} \times \mathcal{B}$, $\Pr[\boldsymbol{\pi}(a, b) = f(a, b)] > 1 - \varepsilon$ (the probability is over the choice of $\boldsymbol{\pi}$).*

We may then define:

- $\mathsf{L}^{\mathrm{A}}_d(f)$ *is the minimum leaf complexity $\mathsf{L}(\pi)$ among all deterministic protocols π that compute f, and have round complexity $\mathsf{R}(\pi) \leq d$, and such that the root node of π is Alice's.* $\mathsf{L}^{\mathrm{B}}_d(f)$ *is defined likewise, but for protocols where the root node is Bob's.*
- $\mathsf{L}^{\mathrm{A}}_{d,\varepsilon}(f)$ *is the minimum $\mathsf{L}(\boldsymbol{\pi})$ among all randomized protocols $\boldsymbol{\pi}$ that compute f with error ε, and have $\mathsf{R}(\boldsymbol{\pi}) \leq d$, and such that the root node of $\boldsymbol{\pi}$ is (always) Alice's.* $\mathsf{L}^{\mathrm{B}}_{d,\varepsilon}(f)$ *is defined likewise for randomized protocols where the root node is (always) Bob's.*
- $\mathsf{C}^{\mathrm{A}}_d$, $\mathsf{C}^{\mathrm{B}}_d$, $\mathsf{C}^{\mathrm{A}}_{d,\varepsilon}$, $\mathsf{C}^{\mathrm{B}}_{d,\varepsilon}$ *are defined analogously where the communication complexity $\mathsf{C}(\pi)$ replaces the leaf complexity $\mathsf{L}(\pi)$.*

Since the communication transcript of a given run of protocol determines the leaf, it must follow that:

▶ **Proposition 7.** *For any protocol π, $\log \mathsf{L}(\pi) \le \mathsf{C}(\pi)$.*

Chromatic number. All our NP-hardness reductions are from the chromatic number problem:

▶ **Definition 8** (Chromatic number). *A* coloring *of an undirected graph G, is a partition of the vertices such that no edge has both endpoints in the same part. The* chromatic number *of a graph G, denoted $\chi(G)$, is the smallest number of parts in a coloring of G.*

The NP-hardness of approximating the chromatic number has been established by a series of results [42, 24, 20], culminating in a paper by Zuckerman [65], where the following was proven:

▶ **Theorem 9** (Hardness For Chromatic Number). *For every $\epsilon > 0$, there is a deterministic polynomial time algorithm that on an input $x \in \{0,1\}^*$ outputs a graph G on n vertices such that*

- *if x is a YES instance of* SAT*, then $\chi(G) \le n^{\epsilon}$, and*
- *if x is a NO instance of* SAT*, then $\chi(G) \ge n^{1-\epsilon}$.*

3 Warmup: deterministic 3-round protocols, large output alphabet

In this section, we show that it is NP-hard to approximate the deterministic 3-round communication complexity of a given matrix over a large alphabet.

We start by observing that deterministic 3-round communication complexity may be approximated by a very simple combinatorial quantity.

▶ **Definition 10.** *Let $\mathcal{A}, \mathcal{B}, \mathcal{Z}$ be finite sets, and let M be an $\mathcal{A} \times \mathcal{B}$ matrix over $\mathcal{Z}$. Let $\mathcal{P} = \{P_i\}_{i \in [k]}$ be a partition of (the columns) $\mathcal{B}$ for some $k \in \mathbb{N}$. (That is, $\emptyset \ne P_i \subseteq \mathcal{B}$, $\bigcup_{i\in[k]} P_i = \mathcal{B}$ and $P_i \cap P_j = \emptyset$ for every $i \ne j$.) For a subset $P \subseteq \mathcal{B}$ of columns, we denote by $\mathsf{Cost}_M(\mathcal{P})$ the number of distinct rows of M restricted to columns in P, i.e.,*

$$\mathsf{Cost}_M(\mathcal{P}) = |\{x_P \in \mathcal{Z}^P \mid x \in \mathcal{Z}^{\mathcal{B}} \text{ is a row of } M\}|.$$

We further define $\mathsf{Cost}_M(\mathcal{P})$ to be $\sum_{i=1}^{k} \mathsf{Cost}_M(P_i)$.

▶ **Proposition 11.** *Let $f \colon \mathcal{A} \times \mathcal{B} \to \mathcal{Z}$ be a function and M be the $\mathcal{A} \times \mathcal{B}$ communication matrix (with entries in $\mathcal{Z}$) that corresponds to f. Let $q \in \mathbb{N}$ be the maximum number of distinct values in a single row of M. We then have*

$$L \le \mathsf{L}_3^{\mathrm{B}}(f) \le L \cdot q, \qquad \text{where } L = \min_{\mathcal{P}} \mathsf{Cost}_M(\mathcal{P}),$$

and where the minimum is taken over all partitions $\mathcal{P}$ of $\mathcal{B}$. Furthermore we have that

$$\log L \le \mathsf{C}_3^{\mathrm{B}}(f) \le \log L + \log q + O(1).$$

Proof. It may be easily seen that $\mathsf{L}_2^{\mathrm{A}}(f)$ is lower-bounded by the number of distinct rows in the communication matrix of f. Because if Alice's partition of the rows includes a part with two different rows, then Bob's ensuing partition of the columns cannot avoid having a non-monochromatic column. It then follows that, if we have a 3-round protocol π where Bob speaks first, and $\mathcal{P}$ is Bob's partition of the columns in the first round, then $\mathsf{Cost}_M(\mathcal{P})$ is a lower-bound on smallest number of leaves that π needs to use to compute f, and thus $\mathsf{L}_3^{\mathrm{B}}(f) \ge L$. The lower-bound on $\mathsf{C}_3^{\mathrm{B}}(f)$ now follows from Proposition 7.

Conversely, it is also easy to see that $\mathsf{L}_2^{\mathrm{A}}(f)$ is upper-bounded by q times the number of distinct rows in the communication matrix of f. The protocol that achieves this bound has Alice tell which kind of row she has, and now in each rectangle the rows are all equal, hence Bob can just tell Alice the color of his column in this row, of which there are q possibilities. Thus $\mathsf{L}_3^{\mathrm{B}}(f) \leq L \cdot q$, and the same protocol shows that $\mathsf{C}_3^{\mathrm{B}}(f) \leq \log L + \log q + O(1)$. ◀

In general, the approximate factor q of Proposition 11 can be as large as $|\mathcal{Z}|$. However, in the following construction, we will construct a matrix where each row has at most $q = 3$ distinct values; in this case, Proposition 11 guarantees that $\min_{\mathcal{P}} \mathsf{Cost}_M(\mathcal{P})$ provides a 3-factor approximation of $\mathsf{L}_3(M)$.

▶ **Theorem 12.** *Given an undirected graph $G = ([n], E)$ with n vertices and $|E| = m > 0$ edges, one may construct in deterministic polynomial time a function $f_G \colon [a] \times [n] \to \{0, 1, \ldots, \ell\}$, with $\ell = m^2 n$, $k = \sqrt{n\ell} = mn$ and $a = \ell + m \cdot k^2$, such that*

$$\chi(G) \cdot \ell \leq \mathsf{L}_3^{\mathrm{B}}(f_G) \leq \chi(G) \cdot 6\ell.$$

Furthermore, we also have

$$\log \chi(G) + \log \ell \leq \mathsf{C}_3^{\mathrm{B}}(f_G) \leq \log \chi(G) + \log \ell + O(1).$$

Proof. We let A_ℓ denote the $\ell \times 1$ column vector,

$$A_\ell = \begin{bmatrix} 1 \\ \vdots \\ \ell \end{bmatrix}.$$

We let B_k and C_k denote the $k^2 \times 1$ column vectors

$$B_k = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ \vdots \\ k \\ \vdots \\ k \end{bmatrix}, \qquad C_k = \begin{bmatrix} 1 \\ \vdots \\ k \\ \vdots \\ 1 \\ \vdots \\ k \end{bmatrix},$$

where each value $i \in [k]$ appears k times. Given an edge $\{v, w\} \in [n] \times [n]$, with $v < w$, we define the $k^2 \times n$ matrix M so that

$$M_{\{v,w\}} = \begin{bmatrix} 0 & \ldots & 0 & B_k & 0 & \ldots & 0 & C_k & 0 & \ldots & 0 \end{bmatrix},$$

where B_k appears in the v-th column and C_k in the w-th column. Finally, let $E = \{e_1, \ldots, e_m\}$ denote the edges of G; then we define the $a \times n$ communication matrix f_G so that

$$f_G = \begin{bmatrix} A_\ell \quad \cdots \quad A_\ell \\ M_{e_1} \\ \vdots \\ M_{e_m} \end{bmatrix} \in \{0, 1, \ldots, \ell\}^{a \times n},$$

where $a = \ell + m \cdot k^2$. Observe that each row of f_G has at most $q = 3$ distinct values; thus, Proposition 11 provides a 3-factor approximation. We now show the stated inequality, namely, that

$$\chi(G) \cdot \ell \leq \mathsf{L}_3^{\mathrm{B}}(f_G) \leq \chi(G) \cdot 2q\ell.$$

The upper-bound is easy to see. Given a coloring of G into $\chi(G)$ colors, we may take the 3-round protocol π where Bob first tells Alice which color his vertex has. This partitions the columns by a partition $\mathcal{P} = \{P_c\}_{c \in [\chi(G)]}$ formed of the various color classes of our coloring. Fix any color c and consider $\mathsf{Cost}_{f_G}(P_c)$. Since P_c is an independent set of G, the C_k and B_k columns of each M_{e_i} sub-matrix will always be placed in different parts; therefore, $\mathsf{Cost}_{f_G}(P_c) \leq \ell + mk$, where the first term counts the number of rows in A_ℓ and the second term counts the number of distinct rows in M_{e_i} restricted to columns of P_c for each $i \in [m]$. Using Proposition 11, we obtain

$$\mathsf{L}_3^{\mathrm{B}}(f_G)/q \leq \mathsf{Cost}_{f_G}(\mathcal{P}) = \sum_c \mathsf{Cost}_{f_G}(P_c) \leq \chi(G) \cdot (\ell + mk) = \chi(G) \cdot 2\ell,$$

and Proposition 11 also gives us

$$\mathsf{C}_3^{\mathrm{B}}(f_G) \leq \log \chi(G) + \log \ell + O(1).$$

For the other direction, let π be any 3-round protocol for f_G. The first round of π partitions the columns of f_G, which is to say, it partitions the vertices of G by some partition $\mathcal{P} = \{P_i\}_{i \in I}$. We claim that $\chi(G) \cdot \ell \leq \mathsf{Cost}_{f_G}(\mathcal{P})$ by analyzing the following two cases.

1. If the partition $\mathcal{P}$ does not form a coloring of G, then there must exist an edge $\{v, w\}$ such that the v-th column and the w-th column of G are placed in the same part P_i. This means that the C_k and B_k columns of the $M_{\{v,w\}}$ sub-matrix are both placed in P_i. In this case, we have $\mathsf{Cost}_{f_G}(P_i) \geq k^2 \geq n \cdot \ell \geq \chi(G) \cdot \ell$, and thus the lower bound holds.
2. Otherwise, suppose that the partition $\mathcal{P}$ does form a coloring of G. Then the number of parts is $\geq \chi(G)$, and so just the contribution from the first ℓ rows gives us $\mathsf{Cost}_{f_G}(\mathcal{P}) \geq \chi(G) \cdot \ell$.

The lower-bounds on $\mathsf{L}_3^{\mathrm{B}}(f_G)$ and $\mathsf{C}_3^{\mathrm{B}}(f_G)$ then follow from Proposition 11. ◀

The following corollary shows that it is NP-hard to approximate $\mathsf{L}_3^{\mathrm{B}}(f)$, for a given total two player function $f : [N] \times [N] \to [N]$, with an approximation ratio better than (roughly) $N^{\frac{1}{5}}$.

▶ **Corollary 13.** *For every $L \subseteq \{0,1\}^*$ in* NP *and every constant $\varepsilon > 0$, there exists a polynomial-time algorithm that, on input $x \in \{0,1\}^*$, outputs a communication matrix $M \in [N]^{N \times N}$ such that if $x \in L$ then $\mathsf{L}_3^{\mathrm{B}}(M) \leq N$, and if $x \notin L$ then $\mathsf{L}_3^{\mathrm{B}}(M) > N^{\frac{6}{5}-\varepsilon}$.*

Proof. We compose the hardness of approximation result for chromatic number in Theorem 9 with the reduction of Theorem 12, padding the communication matrix with all-0 columns to make it square (since Bob speaks first, this adds at most one leaf), so the communication matrix of f is an $N \times N$ matrix with $N = a = \Theta(n^5)$. The parameters then come from the fact that $n = \Theta(N^{\frac{1}{5}})$. ◀

Using the bounds on $\mathsf{C}_3^{\mathrm{B}}$ instead of the bounds on $\mathsf{L}_3^{\mathrm{B}}$ from Theorem 12, we conclude that it is NP-hard to approximate $\mathsf{C}_3^{\mathrm{B}}(f)$, for a given total two player function $f : [N] \times [N] \to [N]$, with an approximation ratio better than (roughly) $\frac{6}{5}$. More precisely:

▶ **Corollary 14.** *For every $L \subseteq \{0,1\}^*$ in* NP *and every $\varepsilon > 0$, there exists a polynomial-time algorithm that, on input $x \in \{0,1\}^*$, outputs a communication matrix $M \in [N]^{N \times N}$ such that if $x \in L$ then $\mathsf{C}_3^{\mathrm{B}}(M) \leq \log N$, and if $x \notin L$ then $\mathsf{C}_3^{\mathrm{B}}(M) > (\frac{6}{5} - \varepsilon) \log N$.*

4 Hardness for deterministic 3-round protocols

Building on the proof ideas presented in Section 3, in this section, we prove NP-hardness of approximating the communication complexity of deterministic 3-round protocols. A key building block is to use the notion of universal set.

▶ **Definition 15** (Universal set). *Let $r, c, k \in \mathbb{N}$, and let $M \in \{0,1\}^{r \times c}$ be a matrix whose columns are $M^{(1)}, \dots, M^{(c)}$. We say that M is (c,k)-*universal *if, for every set $\{y_1, \dots, y_k\} \subseteq [c]$ of k columns of M, the matrix*

$$\begin{bmatrix} M^{(y_1)} & \dots & M^{(y_k)} \end{bmatrix}$$

*has 2^k distinct rows. The set of all the rows of M is called a (c,k)-*universal set*.*

We use the explicit construction of a universal set due to [49].

▶ **Lemma 16** (Naor, Schulman and Srinivasan [51]). *There exists a deterministic algorithm that, given c and $k \in \mathbb{N}$, outputs a (c,k)-universal matrix $M \in \{0,1\}^{r \times c}$ such that $r = 2^{k+O(\log k)^2} \cdot \log c$ in time a polynomial in c and 2^k.*

Now we state the main result of this section.

▶ **Theorem 17.** *Let $\epsilon > 0$ be an arbitrary constant. Given an undirected graph $G = ([n], E)$ with n vertices and $|E| = m > 0$ edges, one may construct in deterministic polynomial time a function $f_G : [a] \times [b] \to \{0,1\}$ and a number $\ell \in \mathbb{N}$, with $a, b, \ell = O(n^8)$, such that*

$$\chi(G) \cdot \ell \le \mathsf{L}_3^{\mathrm{B}}(f_G) \le \chi(G) \cdot \ell^{1+\epsilon}$$

and

$$\log \chi(G) + \log \ell \le \mathsf{C}_3^{\mathrm{B}}(f_G) \le \log \chi(G) + (1+\epsilon) \cdot \log \ell.$$

The idea of the proof is to build upon the reduction in the proof of Theorem 12, by replacing each column with entries from $[\ell]$ with a block of columns that have entries from $\{0,1\}$. The difficulty in making this work is that a partition of the columns might not respect our blocks and could place columns from the same block into different parts. We solve this by thinking as follows. Either the partition is large, meaning it has many parts, so the protocol also has many leaves, which proves our lower bound, or otherwise for any block C_v of columns the part P_i which has most columns from C_v has many columns from C_v; we may then act as if "C_v was placed in P_i".

Proof. Let $t, k, c \in \mathbb{N}$ be parameters chosen later. Let $A \in \{0,1\}^{r \times c}$ and $M \in \{0,1\}^{s \times c}$ be the (c,t)-universal and (c,k)-universal matrices, respectively, that are constructed by the polynomial-time deterministic algorithm of Lemma 16; then we have $r = 2^{(1+o(1)) \cdot t} \cdot \log c$ and $s = 2^{(1+o(1)) \cdot k} \cdot \log c$.

Let $x_1, \dots, x_s$ be the rows of M. We then let B and C denote the $s^2 \times c$ matrices

$$B = \begin{bmatrix} x_1 \\ \vdots \\ x_1 \\ \vdots \\ x_s \\ \vdots \\ x_s \end{bmatrix}, \qquad C = \begin{bmatrix} x_1 \\ \vdots \\ x_s \\ \vdots \\ x_1 \\ \vdots \\ x_s \end{bmatrix},$$

where a row x_i appears s times for each $i \in [s]$. Given an edge $\{v, w\} \in [n] \times [n]$, with $v < w$, we define the $s^2 \times c \cdot n$ matrix:

$$M_{\{v,w\}} = \begin{bmatrix} 0 & \dots & 0 & B & 0 & \dots & 0 & C & 0 & \dots & 0 \end{bmatrix},$$

where B appears in the v-th block of c columns and C in the w-th block of c columns. Finally, let $E = \{e_1, \dots, e_m\}$ denote the edges of G; then we define the $(r + m \cdot s^2) \times c \cdot n$ communication matrix f_G so that

$$f_G = \begin{bmatrix} A & \dots & A \\ & M_{e_1} & \\ & \vdots & \\ & M_{e_m} & \end{bmatrix} \in \{0,1\}^{(r+m\cdot s^2)\times c\cdot n}.$$

Let $\ell := 2^t$. Let $\mathcal{P}$ be a partition of the columns of f_G that minimizes $\mathsf{Cost}_{f_G}(\mathcal{P})$. We now show that, for a suitable choice of t, k and c,

$$\chi(G) \cdot \ell \le \mathsf{Cost}_{f_G}(\mathcal{P}) \le \chi(G) \cdot \ell^{1+o(1)}.$$

This will complete the proof, because Proposition 11 shows that, for a binary matrix f_G, $\mathsf{Cost}_{f_G}(\mathcal{P})$ is a 2-factor approximation of $\mathsf{L}_3^{\mathrm{B}}(f_G)$ and $\log \mathsf{Cost}_{f_G}(\mathcal{P})$ is an approximation of $\mathsf{C}_3^{\mathrm{B}}(f_G)$ up to an additive $O(1)$ term. We will choose the parameters t, k and c so that the following conditions are satisfied.

1. Condition 1. $r + ms \le \ell^{1+o(1)}$.
2. Condition 2. $c \ge n\ell \cdot \max\{t, k\}$.
3. Condition 3. $2^{2k} \ge n\ell$.

Given that Condition 1 is satisfied, the complexity upper-bounds are easy to see. Let $\mathcal{P}$ be a coloring of G that partitions the vertex set into $\chi(G)$ parts. Since no class contains an edge, the C and B sub-matrices of each M_{e_i} sub-matrix will always be placed in different parts, and we thus have

$$\mathsf{Cost}(\mathcal{P}) \le \chi(G) \cdot (r + ms) \le \chi(G) \cdot \ell^{1+o(1)}.$$

For the other direction, we will lower-bound $\mathsf{Cost}(\mathcal{P})$ for every partition $\mathcal{P}$. To begin, if the number of parts is $|I| \ge n\ell$, then we must conclude that $\mathsf{Cost}(\mathcal{P}) \ge n\ell \ge \chi(G) \cdot \ell$, as desired.

Otherwise, $|I| \le n\ell$. Then for every block of columns $v \in [n]$ there must exist a part $P_{i(v)}$ which contains at least $c/|I| \ge c/n\ell$ columns from the v-th block. Observe that $c/n\ell \ge \max\{t, k\}$ by Condition 2.

Now, either the mapping $v \mapsto i(v)$ is a valid coloring of G or not. First, suppose that the mapping $v \mapsto i(v)$ is not a valid coloring of G, meaning that there exists an edge $\{v, w\} \in E$ such that $i = i(v) = i(w)$. Then the v-th column block and the w-th column block each have at least $\max\{t, k\}$ columns in the same part P_i. This will mean that the C and B sub-matrices of the $M_{\{v,w\}}$ sub-matrix each have at least k columns in P_i. But then, since M is (c, k)-universal, it follows from Condition 3 that $\mathsf{Cost}(\mathcal{P}) \ge \mathsf{Cost}(P_i) \ge 2^{2k} \ge n\ell$.

Next, suppose that $i\colon [n] \to \mathcal{P}$ does form a coloring of G. Then there exist at least $\chi(G)$ parts P_i each receiving at least t columns from some column block, and so, since A is (c, t)-universal, the contribution from the A columns give us $\mathsf{Cost}(\mathcal{P}) \ge \chi(G) \cdot 2^t = \chi(G) \cdot \ell$.

This will give us the theorem, and all we are left to do is ensure that the various conditions can be met: We define t so that $\ell = 2^t$ satisfies that $\ell \leq (nm^2)^{1+\epsilon/2} < 2\ell$. To meet Condition 3, let k be the smallest integer satisfying that $n\ell \leq 2^{2k}$. Let $c := n\ell \cdot \max\{t, k\}$ so that Condition 2 is satisfied. Finally, observe that Condition 1 is satisfied because

$$r + ms \leq \ell^{1+o(1)} \cdot \log c + m \cdot 2^{(1+o(1))\cdot k} \cdot \log c \leq \ell^{1+o(1)} + m(n\ell)^{\frac{1}{2}+o(1)} \cdot \log c \leq \ell^{1+\epsilon},$$

where the last inequality holds for all large n, m. ◀

We can now prove that $\mathsf{L}_3^{\mathrm{A}}$ and $\mathsf{C}_3^{\mathrm{A}}$ are hard to approximate, also for Boolean functions.

▶ **Theorem 18.** *For every constant $\varepsilon > 0$, there exists a deterministic quasipolynomial-time algorithm that, on input $x \in \{0,1\}^*$, outputs a communication matrix $M \in \{0,1\}^{N\times N}$ and a number $k \in \mathbb{N}$, with $k \leq N = |x|^{O(1)}$, such that*

1. *if x is a YES instance of* SAT*, then $\mathsf{L}_3^{\mathrm{B}}(M) \leq k$ and $\mathsf{C}_3^{\mathrm{B}}(M) \leq \log k$, and*
2. *if x is a NO instance of* SAT*, then $\mathsf{L}_3^{\mathrm{B}}(M) \geq N^{\frac{1}{8}-\varepsilon} \cdot k$ and $\mathsf{C}_3^{\mathrm{B}}(M) \geq \log k + (\frac{1}{8} - \varepsilon)\log N$.*

Proof. We combine the two reductions of Theorems 9 and 17, which we invoke with the same parameter ε. Fix any input x and let G be an n-vertex graph that is produced by the reduction of Theorem 9 on input x. Let $M \in \{0,1\}^{a\times b}$ be the communication matrix of f_G, and $\ell \in \mathbb{N}$, be given by the reduction of Theorem 17 on input G, and set $N = \max(a, b) = O(n^8)$, $k = n^\varepsilon \cdot \ell^{1+\varepsilon}$. We may assume that $a = b$ without loss of generality, since otherwise we may pad M with all-0 rows or all-0 columns, and this changes the leaf complexity by at most a factor of 2, and the communication complexity by at most 1 bit, so this change makes no difference to the result.

We verify that the inequalities are satisfied for M: If $x \in L$, then we have $\mathsf{L}_3^{\mathrm{B}}(M) \leq \chi(G) \cdot \ell^{1+\epsilon} \leq n^\varepsilon \cdot \ell^{1+\epsilon} = k$. If $x \notin L$, then we have $\mathsf{L}_3^{\mathrm{B}}(M) \geq \chi(G) \cdot \ell \geq n^{1-\epsilon} \cdot \ell \geq N^{\frac{1}{8}-O(\epsilon)} \cdot k$. Similar inequalities hold for $\mathsf{C}_3^{\mathrm{B}}(M)$. Since ε can be arbitrarily small, we may ignore the constant factors. ◀

5 From 3-rounds to multiple rounds using deterministic round elimination

In this section we show how we can use an oracle for computing $\mathsf{L}_{d+1}^{\mathrm{B}}$ in order to compute $\mathsf{L}_d^{\mathrm{A}}$. The gadget in our reduction involves a special function, XorEq_k, which is a small modification of the standard Equality function.

▶ **Definition 19.** *The function $\mathsf{XorEq}_k : [k] \times ([k] \times \{0,1\}) \to \{0,1\}$ is given by*

$$\mathsf{XorEq}_k(x; y, z) = \begin{cases} z & \text{if } x \neq y. \\ 1 - z & \text{if } x = y. \end{cases}$$

The key property of XorEq_k is that it has a fooling set lower bound that is tight. In particular, $\{(x; y, z) : x = y\}$ is a fooling set of cardinality $2k$, and there is a 2-round protocol for solving XorEq_k with $2k$ leaves (where Alice just sends her full input to Bob, and he replies with the output).

▶ **Lemma 20** (Fooling set lower-bound for XorEq_k)**.** *Let π be a protocol for solving the function $f : ([k] \times [a]) \times ([k] \times \{0,1\}) \to \{0,1\}$ given by $f(x_0, x_1; y, z) = \mathsf{XorEq}_k(x_0; y, z)$.*[3] *Then*

$$\pi(x_0, x_1; y, z) \neq \pi(x_0', x_1'; y', z')$$

if either

- $y = x_0 \neq x_0'$, *or*
- $y = x_0 = x_0'$ *and* $z \neq z'$.

Proof. First, suppose for contradiction that $\pi(x_0, x_1; y, z) = \pi(x_0', x_1'; y', z')$ and $y = x_0 \neq x_0'$. Since leaves are combinatorial rectangles, we can infer that $\pi(x_0', x_1'; y, z) = \pi(x_0, x_1; y, z)$. But since $y = x_0 \neq x_0'$, we know that

$$f(x_0, x_1; y, z) = 1 - z \neq z = f(x_0', x_1'; y, z)$$

so this contradicts that $\pi(x_0, x_1; y, z)$ is a monochromatic leaf.

Similarly, if $y = x_0 = x_0'$ and $z \neq z'$, then we have $f(x_0, x_1; y, z) \neq f(x_0', x_1'; y', z')$, so $\pi(x_0, x_1; y, z) \neq \pi(x_0', x_1'; y', z')$ by monochromaticness. ◀

The main technical portion of our reduction is the following *deterministic* variant of the round-elimination lemma.

▶ **Lemma 21** (Restatement of Lemma 3)**.** *Let $d \geq 3$. Let $f : [a] \times [b] \to \{0,1\}$. Let $F : ([k] \times [a]) \times ([k] \times [b] \times \{0,1\} \times \{0,1\}) \to \{0,1\}$ be given by*

$$F(x_0, x_1; y_0, y_1, z, i) = \begin{cases} \mathsf{XorEq}_k(x_0; y_0, z) & , \text{ if } i = 0 \\ f(x_1; y_1) & , \text{ if } i = 1. \end{cases}$$

Then we have

$$\min\{4k, 2k - 2 + \mathsf{L}_d^{\mathrm{B}}(f)\} \leq \mathsf{L}_{d+1}^{\mathrm{A}}(F) \leq 2k + \mathsf{L}_d^{\mathrm{B}}(f).$$

Proof. The upper bound comes from the protocol where Alice skips the first round of communication, Bob sends i to Alice and begins running the best d-round Bob-first protocol for XorEq_k or f, based on whether $i = 0$ or $i = 1$. In particular, we have that

$$\mathsf{L}_{d+1}^{\mathrm{A}}(F) \leq \mathsf{L}_d^{\mathrm{B}}(\mathsf{XorEq}_k) + \mathsf{L}_d^{\mathrm{B}}(f) \leq 2k + \mathsf{L}_d^{\mathrm{B}}(f),$$

where the last upper bound uses that $d \geq 3$.

We now argue the lower bound. Suppose π is a $(d+1)$-round Alice-first protocol for F. Let $\mathcal{L} = \{\ell_1, \ldots, \ell_{\mathsf{L}(\pi)}\}$ denote the set of leaves of π.

Our arguments split into two cases depending on whether there is a *good* input $x_0 \in [k]$. We say an input $x_0 \in [k]$ is good if all of Alice's inputs that begin with x_0 are placed in a single partition. More formally, let $\mathcal{P} = \{P_1, \ldots, P_{|\mathcal{P}|}\}$ be the partition of Alice's inputs corresponding to the first round of π. We say $x_0 \in [k]$ is *good* if there exists a $q \in [|\mathcal{P}|]$ such that $\{x_0\} \times [a] \subseteq P_q$.

[3] In our definition of f, the input x_1 does not affect the output of the function. The fact that this lemma holds even when there is an extraneous input like x_1 will be used later.

Case 1: There is a good input. Suppose that there exists a good input $x_0^\star \in [k]$ such that $\{x_0^\star\} \times [a] \subseteq P_q$ for some q. Let $\mathcal{L}_{x_0^\star}$ be the set of leaves of π that contain an input where $x_0 = x_0^\star$, that is,

$$\mathcal{L}_{x_0^\star} = \{\ell \in \mathcal{L} : \exists (x_1, y_0, y_1, z, i) \text{ with } \pi(x_0^*, x_1; y_0, y_1, z, i) = \ell\}$$

and let $\overline{\mathcal{L}_{x_0^\star}}$ denote the complementary set of leaves, that is $\overline{\mathcal{L}_{x_0^\star}} = \mathcal{L} \setminus \mathcal{L}_{x_0^\star}$. We will show that $|\mathcal{L}_{x_0^\star}| \geq \mathsf{L}_d^{\mathrm{B}}(f)$ and that $|\overline{\mathcal{L}_{x_0^\star}}| \geq 2k-2$. As a result, we get that

$$\mathsf{L}(\pi) \geq |\mathcal{L}_{x_0^\star}| + \overline{\mathcal{L}_{x_0^\star}} \geq \mathsf{L}_d^{\mathrm{B}}(f) + 2k - 2,$$

as desired.

First, we show that $|\mathcal{L}_{x_0^\star}| \geq \mathsf{L}_d^{\mathrm{B}}(f)$. Let π' be the d-round Bob-first subprotocol of π given when Alice says that her input is in P_q at the first round. Since $\{x_0^\star\} \times [a] \subseteq P_q$, it follows that π' computes $f(x_1, y_1)$ on input $(x_0^\star, x_1; y_0, y_1, 0, 1)$ for all $x_1 \in [a]$ and $y_1 \in [b]$. This yields a d-round Bob-first protocol for computing f, and therefore, the number of leaves in π' containing the input $x_0^\star$ must be at least $\mathsf{L}_d^{\mathrm{B}}(f)$. Hence, $|\mathcal{L}_{x_0^\star}| \geq \mathsf{L}_d^{\mathrm{B}}(f)$.

Next, we argue that $|\overline{\mathcal{L}_{x_0^\star}}| \geq 2k-2$. Consider the set of leaves given by

$$\{\pi(x_0, x_1; y_0, y_1, z, i) : i = 0, x_0 = y_0 \in [k] \setminus \{x_0^\star\}, y_1 = 1, z \in \{0, 1\}\}.$$

If we consider the restriction of π to the inputs $y_1 = 1$ and $i = 0$, we can apply Lemma 20 to conclude that all $2k-2$ leaves in this set are in $\overline{\mathcal{L}_{x_0^\star}}$ and are pairwise distinct.

Case 2: No good input. Now we consider the case where there is no good input. For each $x_0 \in [k]$, we define a set $A_{x_0} \subseteq [a]$ of cardinality 2 as follows. Since $x_0 \in [k]$ is not good, there exists a pair $(x_1, x_1') \in [a]^2$ such that $\pi(x_0, x_1; y_0, y_1, z, i) \neq \pi(x_0, x_1'; y_0, y_1, z, i)$ for all y_0, y_1, z and i. Let $A_{x_0} = \{x_1, x_1'\}$. This completes our definition of A_{x_0}.

We claim that the $4k$ inputs in the following set are all in pairwise distinct leaves:

$$W = \{(x_0, x_1, y_0, y_1, z, i) : x_0 = y_0 \in [k], i = 0, x_1 \in A_{x_0}, y_1 = 1, z \in \{0, 1\}\}.$$

To see this, suppose that $w \neq w'$ for some $w, w' \in W$. Let $w = (x_0, x_1; y_0, y_1, z, i)$ and $w' = (x_0', x_1'; y_0', y_1', z', i')$. We prove $\pi(w) \neq \pi(w')$ by considering the following three cases.

1. If $x_0 = x_0'$ and $x_1 \neq x_1'$, then we know that $\{x_1, x_1'\} = A_{x_0}$. By the construction of A_{x_0}, we can conclude that $\pi(w) \neq \pi(w')$.
2. If $x_0 = x_0'$ and $x_1 = x_1'$, then we must have $x_0 = y_0 = x_0'$ and $z \neq z'$ since $w \neq w'$; using Lemma 20, we conclude that $\pi(w) \neq \pi(w')$.
3. Lastly, suppose that $x_0 \neq x_0'$. If we consider the restriction of π to the inputs $y_1 = 1$ and $i = 0$, we can apply Lemma 20 to conclude that $\pi(w) \neq \pi(w')$. ◀

Using this lower bound, we show one can approximately compute round-d complexity given an oracle that approximately computes round-$(d+1)$ complexity. We consider the following notion of approximation.

▶ **Definition 22.** *For every constant $\epsilon > 0$, we say that an oracle $\mathcal{O}$ is a $(1+\epsilon)$-approximation of a function $\mathcal{L}(\cdot)$ if there exists a constant c such that, for all g in the domain of $\mathcal{L}(\cdot)$,*

$$\mathcal{L}(g) \leq \mathcal{O}(g) \leq (1+\epsilon) \cdot \mathcal{L}(g) + c.$$

▶ **Corollary 23.** *Let $0 < \epsilon < \frac{1}{8}$. Given an oracle computing a $(1+\epsilon)$-approximation of $\mathsf{L}_{d+1}^{\mathrm{A}}(\cdot)$ and a function $f : [a] \times [b] \to \{0, 1\}$, one can deterministically compute a $(1+4\epsilon)$-approximation of $\mathsf{L}_d^{\mathrm{B}}(f)$ in time $(ab)^{O(1)}$*

Proof. First, we give the reduction algorithm and then we prove its correctness. Suppose $\mathcal{O}$ is an oracle that computes an approximation of $\mathsf{L}^{\mathrm{A}}_{d+1}(\cdot)$ satisifying, for all functions g,

$$\mathsf{L}^{\mathrm{A}}_{d+1}(g) \le \mathcal{O}(g) \le (1+\epsilon)\mathsf{L}^{\mathrm{A}}_{d+1}(g) + O(1).$$

For a positive integer k, let $F_k : ([k] \times [a]) \times ([k] \times [b] \times \{0,1\} \times \{0,1\}) \to \{0,1\}$ be given by

$$F_k(x_0, x_1; y_0, y_1, z, i) = \begin{cases} \mathsf{XorEq}_k(x_0; y_0, z) & , \text{ if } i = 0 \\ f(x_1; y_1) & , \text{ if } i = 1. \end{cases}$$

The reduction computes

$$v := \max\{\mathcal{O}(F_k) - 2(1+\epsilon)k : k \in [ab]\},$$

and outputs $v' := (v+2)/(1-2\epsilon)$. It is easy to see that this reduction runs in time $(ab)^{O(1)}$.

To prove the correctness of the reduction, we claim that

$$\mathsf{L}^{\mathrm{B}}_{d}(f) \le v' \le (1+4\epsilon) \cdot \mathsf{L}^{\mathrm{B}}_{d}(f) + O(1)$$

for all functions f.

From Lemma 21, we know that for all k

$$\mathcal{O}(F_k) - 2(1+\epsilon)k \le (1+\epsilon)\mathsf{L}^{\mathrm{A}}_{d+1}(F_k) - 2(1+\epsilon)k + O(1) \le (1+\epsilon)\mathsf{L}^{\mathrm{B}}_{d}(f) + O(1)$$

so we have that

$$v' = \frac{v+2}{1-2\epsilon} \le \frac{1+\epsilon}{1-2\epsilon} \cdot \mathsf{L}^{\mathrm{B}}_{d}(f) + O(1) \le (1+4\epsilon) \cdot \mathsf{L}^{\mathrm{B}}_{d}(f) + O(1),$$

where the last inequality holds because $\epsilon < 1/8$.

On the other hand, if $k = \mathsf{L}^{\mathrm{B}}_{d}(f)$, we have from Lemma 21 that

$$\begin{aligned} \mathcal{O}(F_k) - 2(1+\epsilon)k &\ge \mathsf{L}^{\mathrm{A}}_{d+1}(F_k) - 2(1+\epsilon)k \\ &\ge \min\{4k, 2k - 2 + \mathsf{L}^{\mathrm{B}}_{d}(f)\} - 2(1+\epsilon)k \\ &= \min\{4\mathsf{L}^{\mathrm{B}}_{d}(f), 3\mathsf{L}^{\mathrm{B}}_{d}(f) - 2\} - 2(1+\epsilon)\mathsf{L}^{\mathrm{B}}_{d}(f) \\ &= 3\mathsf{L}^{\mathrm{B}}_{d}(f) - 2 - 2(1+\epsilon)\mathsf{L}^{\mathrm{B}}_{d}(f) \\ &\ge (1-2\epsilon)\mathsf{L}^{\mathrm{B}}_{d}(F_k) - 2. \end{aligned}$$

Since $k = \mathsf{L}^{\mathrm{B}}_{d}(f) \le ab$, we conclude that $v' = (v+2)/(1-2\epsilon) \ge \mathsf{L}^{\mathrm{B}}_{d}(F_k)$. ◀

Combining Corollary 23 with the hardness result for the $d = 3$ case in Theorem 18, we get that computing L_d is NP-hard (under a polynomial-time truth-table reduction).

▶ **Corollary 24.** *For any integer $d \ge 3$, there exists an $\epsilon > 0$ such that given access to an oracle that computes a $(1+\epsilon)$-approximation of $\mathsf{L}^{\mathrm{A}}_{d}$, one can compute any language in NP in polynomial time.*

6 Hardness for randomized 3-round protocols

In this section we prove that it is NP-hard to distinguish whether a function having short deterministic 3-round communication protocols, from a function needs long randomized 3-round protocols with a small error.

▶ **Definition 25.** *The (normalized) Hamming distance of two strings $m_1, m_2 \in \{0,1\}^c$, denoted $\Delta(m_1, m_2)$ is the fraction of bit-positions where m_1 and m_2 differ.*

▶ **Definition 26.** *Let $r, c \in \mathbb{N}$, and let $M \in \{0,1\}^{r \times c}$ be a matrix whose columns are $M^{(1)}, \ldots, M^{(c)}$. Then M is called a (t, k, ε)-gadget if for every set $S = \{s_1, \ldots, s_t\} \subseteq [c]$ of t (distinct) columns of M, the matrix*

$$M_S = \begin{bmatrix} M^{(s_1)} & \ldots & M^{(s_t)} \end{bmatrix}$$

has at least k rows which are pairwise ε-far in Hamming distance. Meaning, there are k rows $m_1, \ldots, m_k \in \{0,1\}^t$ of M_S such that $\Delta(m_i, m_j) \geq \varepsilon$ for all $i, j \in [k]$, $i \neq j$.

▶ **Lemma 27.** *Let $10 \log r \leq c \ll 2^{\frac{r}{10}}$; then a uniformly random matrix $M \in \{0,1\}^{r \times c}$ is a $(10 \log r, \frac{2}{3} r, 1/10)$-gadget.*

Proof. The proof is a standard use of the probabilistic method [13], but let us check the parameters. We choose each entry of M uniformly at random, and we wish to prove that M is a $(t, k, 1/10)$-gadget with high probability. Fix any set S of $t = 10 \log r$ columns – there are $\binom{c}{t}$ many such sets. Imagine we choose r rows sequentially, uniformly at random. Whenever we pick a new row, we call "good" row if it is $\frac{1}{10}$-far, in Hamming distance, of any of the previously picked rows. If this does not hold, we call the row "bad". We wish to upper-bound the probability of seeing fewer than $k = \frac{2}{3} r$ good rows. Notice that the number of t-bit strings in the $\frac{1}{10}$-Hamming-ball around the rows we have seen so far, is less than

$$p = r \binom{t}{\leq t/10} 2^{\frac{t}{10}} \leq r \cdot 2^{(H_2(\frac{1}{10}) + \frac{1}{10}) \cdot t} \leq r \cdot 2^{0.57 \cdot t} \leq r^7,$$

where $H_2(p)$ is the binary entropy function. So the probability of seeing another bad row is less than $p \cdot 2^{-t} \leq r^{-3}$. Hence, the probability of seeing more than $\frac{r}{3}$ bad rows, is less than $(r^{-3})^{\frac{r}{3}} = r^{-r}$. It then follows by a union bound that M will fail to be a $(t, k, \frac{1}{10})$-gadget, with probability no greater than

$$\binom{c}{t} r^{-r}$$

which is close to 0 provided that $c \ll 2^{\frac{r}{10}}$. ◀

We now make the observation that it is possible for a constant-depth to decide, in an approximate sense, whether a given matrix is a good enough gadget. Since there exists an explicit pseudorandom generator for AC_0 with polylog seed length [52], a good gadget can be found in deterministic quasi-polynomial time.

▶ **Corollary 28.** *Let $10 \log r \leq c \ll 2^{\frac{r}{11}}$. One can obtain a matrix $M \in \{0,1\}^{r \times c}$ which is a $(10 \log r, \frac{2}{3} r, 1/10)$-gadget, via a deterministic algorithm running in time $2^{\mathsf{polylog}(r \cdot c)}$.*

Proof. We will show that there exists a constant-depth circuit C of size quasipolynomial in r and c, having $r \times c$ inputs, with the property that every matrix $M \in \{0,1\}^{r \times c}$ with $C(M) = 1$ is a $(10 \log r, \frac{2}{3} r, 1/10)$-gadget, and such that $\Pr[C(M) = 1] = 1 - o(1)$. Corollary 28 follows because there exists an explicit pseudorandom generator $G = \{G_{rc} \colon \{0,1\}^{(\log rc)^{O(1)}} \to \{0,1\}^{r \times c}\}_{r,c \in \mathbb{N}}$ for AC_0 [52].

The circuit checks that every set S of $t = 10 \log r$ columns has at least $\frac{2}{3} r$ good rows, in the same sense as described in the proof of Lemma 27. This is strong enough to ensure that the input is a $(t, k, \frac{1}{10})$-gadget, and it suffices to present a quasipolynomial-size circuit to check this property, since the number of such sets S is itself quasipolynomial.

We cannot check this property exactly, but we can check this property approximately. Using approximate counting [2, 3], a polynomial-size constant-depth circuit D may, given two strings $x, y \in \{0,1\}^t$, give us $D(x,y) = 1$ if $\Delta(x,y) \geq \frac{1}{10} + \varepsilon$, and $D(x,y) = 0$ if $\Delta(x,y) \leq \frac{1}{10}$, where $\varepsilon > 0$ can be chosen to be any arbitrarily small constant. A row x_i will be called *good* if $D(x_i, x_j) = 1$ for all previous rows x_j with $j < i$. Again using approximate counting, and letting M_S denote the sub-matrix of M restricted to the columns in S, we may construct a circuit T_S with $T_S(M) = 1$ if at least $(\frac{2}{3} + \varepsilon)r$ of the rows of M_S are good, and with $T_S(M) = 0$ if fewer than $\frac{2}{3}r$ of the rows of M_S are good. The extra ε will still allow for the previous probability bounds to hold, and if $T_S(M) = 1$ for all S, then M is a $(t, k, \frac{1}{10})$-gadget. ◀

We now show that a simple lower bound on two round communication complexity.

▶ **Lemma 29.** *If $M \in \{0,1\}^{r \times c}$ is a matrix containing r rows $x_1, \ldots, x_r \in \{0,1\}^c$, all pairs of which are ε-far in Hamming distance, then for $\delta = \frac{\varepsilon}{8}$*

$$\mathsf{L}^{\mathrm{A}}_{2,\delta}(M) \geq \frac{1}{2}r,$$

and the hard distribution witnessing this is uniform over the rows and columns of M.

Proof. Let us take an arbitrary matrix $M \in \{0,1\}^{r \times c}$, and think about its $\mathsf{L}^{\mathrm{B}}_1$-complexity. We first observe that the $\mathsf{L}^{\mathrm{B}}_1$-protocol for M which has the smallest possible error is the single-bit protocol where Bob sends the majority of his column to Alice, meaning, he sends 1 if and only if half or more of the entries in his column are 1. Hence, the smallest error which a deterministic $\mathsf{L}^{\mathrm{B}}_1$-protocol can make when computing M under the uniform distribution is precisely the error of this smallest-error protocol, which is

$$\mathsf{Err}(M) = \min_{z \in \{0,1\}^c} \frac{1}{r} \sum_{i=1}^{r} \Delta(x_i, z).$$

Indeed, the z giving the minimum is the column-wise majority of M.

Now, suppose $\Delta(x_i, x_j) \geq \varepsilon$ for all $i, j \in [r]$, $i \neq j$. If we have an $\mathsf{L}^{\mathrm{A}}_2$-protocol π for M which partitions the rows of F into fewer than $\frac{r}{2}$ parts, then there must exist $\frac{r}{2}$ rows of M which get placed in a part that contains at least one other row of M. If we have a part which has p rows placed together, then by the triangle inequality this means that, for any $z \in \{0,1\}^c$, we must have $\Delta(x_i, z) \geq \frac{\varepsilon}{2}$ for at least $p-1$ values of i (if z is $\frac{\varepsilon}{2}$-close to one of the x_i, it must be $\frac{\varepsilon}{2}$-far from all other x_i in the same part, since they are pairwise distant). Hence if M' is any part of M in the partition, having more than one row, we have

$$\mathsf{Err}(M') \geq \frac{p-1}{p}\frac{\varepsilon}{2} \geq \frac{\varepsilon}{4}$$

But since 1/2 of the rows get placed together with other rows, the total error incurred by π on M is at least $\frac{\varepsilon}{8}$. ◀

We now show the following hardness result.

▶ **Theorem 30.** *Let $0 < \delta < 1$ be given. Given an undirected graph $G = ([n], E)$ with n vertices and $|E| = m > 0$ edges, one may construct in deterministic quasipolynomial time a function $f_G : [a] \times [b] \to \{0,1\}$ and a number $\ell \in \mathbb{N}$, with $a, b, \ell = O(n^{27})$, such that*

- $\mathsf{L}^{\mathrm{B}}_{3,n^{-\delta}}(f_G) = \Omega(n^{\frac{\delta}{16}-1} \cdot \chi(G) \cdot \ell)$,
- $\mathsf{C}^{\mathrm{B}}_{3,n^{-\delta}}(f_G) \geq \frac{\delta}{16}\log n + \log\chi(G) - \log n + \log \ell - O(1)$,
- $\mathsf{L}^{\mathrm{B}}_3(f_G) = O(\chi(G) \cdot \ell)$, *and*
- $\mathsf{C}^{\mathrm{B}}_3(f_G) = \log\chi(G) + \log \ell + O(1)$.

Proof. The construction is the same as in Theorem 17, but where use (t,k,ε)-gadgets instead of universal sets, and Lemma 29 instead of Proposition 11. The function f_G is defined exactly as in the proof of Theorem 17, but we use the gadgets from Corollary 28, namely, we set $\ell = m^4n^4$ and $c = n^2\ell^2$, and we choose $A \in \{0,1\}^{r\times c}$ to be an $(O(\log r), \frac{2}{3}r, 1/10)$-gadget, and we choose $B, C \in \{0,1\}^{s\times c}$ to be an $(O(\log s), \frac{2}{3}s, 1/10)$, where $r = \ell = m^4n^4$ and $s = m^3n^3$. This choice obeys the two conditions:

1. Condition 1. $r + ms = O(\ell)$.
2. Condition 2. $\sqrt{c/2} \geq n \cdot \ell$.

The upper-bound is given by the same protocol as in the proof of Theorem 17, where Condition 1 gives us improved parameters. We are left to prove the lower-bound. This is proven via Yao's principle. The hard distribution μ for f_G is chosen as follows:

- With probability 1/2 we will let the input (x,y) be a uniformly chosen row x among the first r rows, and a uniformly chosen column. I.e. a uniform entry of the $[A \dots A]$ sub-matrix of f_G.
- And with probability 1/2 we choose an edge $\{v,w\}$ uniformly at random from the edges of G, and then choose a uniform entry among the B and C sub-matrices of the $M_{\{u,v\}}$ sub-matrix of f_G.

Now suppose we are given a deterministic $\mathsf{L}_3^{\mathsf{B}}$-protocol π with L leaves, which computes f_G with error $\leq n^{-\delta}$ under the distribution μ. We will then show that one of two things must happen: either (1) π has $L \gg n\ell$ leaves, or (2) π gives us a coloring of G with $\leq \frac{3}{\ell}n^{1-\frac{1}{16}\delta}L$ colors. In both cases it must follow that $L = \Omega(n^{\frac{\delta}{16}-1} \cdot \chi(G) \cdot \ell)$.

Indeed, we will show that either (1) π has $L \gg n\ell$ leaves, or (2') π gives us a coloring of a graph G', which has $\leq \frac{1}{\ell}L$ colors, where G' is obtained from G by removing $\leq n^{-\frac{1}{8}\delta}|E|$ edges. We will then make use of the following:

▷ Claim 31. If G' is obtained from G by removing $\leq n^{-\delta} \cdot |E|$ edges, then any coloring of G' with C colors will induce a coloring of G with $3n^{1-\frac{\delta}{2}}C$ colors.

Proof. Let $N = n^{-\delta/2} \cdot n$. Split the vertices of G into two sets: V_1 contains those vertices of G where we have removed $\geq N$ edges, and V_2 contains the remaining vertices. We have that $|V_1| \leq 2N$, otherwise too many edges would have been removed.

So let $\alpha' : [n] \to [C]$ be a C-coloring of G'. We then construct a coloring $\alpha : [n] \to [2N + C \cdot (N+1)]$, as follows. We first color each vertex of V_1 by its own color. Then we greedily color each vertex $v \in V_2$ by a color $\alpha(v) = (\alpha'(v), \beta(v))$, such that the second coordinate $\beta(v) \in [N+1]$ does not appear as the second coordinate of any neighbours $w \in V_2$ of v which we have already colored. There will always exist such a β because the number of new neighbors of v, when going from G' to G, is $\leq N$. This is a coloring of G with $\leq 3C \cdot N$ colors, as promised. ◁

Now, look at the marginal distribution of μ over the columns of f_G. Then each block of columns of f_G corresponding to a vertex v gets probability mass exactly:

$$\frac{1}{2n} + \frac{1}{2}\frac{\deg(v)}{2|E|}. \qquad (*)$$

Let us now remove *high-error columns*, as follows. We first remove from f_G all column blocks v where the error probability of π, conditioned on Bob's input being a column of v, is greater than $n^{-\delta/2}$. Since the error probability of π is $\leq n^{-\delta}$, then by Markov's inequality, the total probability mass removed in this way is less than $n^{-\delta/2}$. By $(*)$, removing all such vertices v from G will remove fewer than $n^{-\delta/2} \cdot 4|E| \leq 4n^{-\delta/2}n$ edges in total.

Now we do similarly inside each block. For each surviving column block v we know that the error probability inside it (i.e. conditioned on getting a column inside the block) is $\leq n^{-\delta/2}$. Let us then remove every column y where the error probability of π, conditioned on Bob's input being y, is greater than $2n^{-\delta/2}$. Notice that, within each block, every column gets the same probability. Hence, again by Markov's inequality, by removing these high-error y we have removed fewer than $\frac{1}{2}c$ columns from each block. We are left with a sub-matrix of f_G where, in each column, π has error probability less than $n^{-\delta/4}$, and where each surviving column block v has $\geq \frac{1}{2}c$ columns.

We now remove some further columns, which we will call *leftover columns*. To begin, we remove enough columns from each surviving block so that there are exactly $\frac{1}{2}c$ columns in each block. We do this so we don't have to think about having a different number of surviving columns among different blocks.

Now let $\mathcal{P} = \{P_1, \ldots, P_{|\mathcal{P}|}\}$ be the partition of the surviving columns of f_G which is induced by the first round of π. If $|\mathcal{P}| \geq \sqrt{c/2} \gg n\ell$ (by Condition 2), then we have established (1). Otherwise, we must show (2'). If $|\mathcal{P}| \leq \sqrt{c/2}$, then for each column block v there exists a part $P_{i(v)}$ of $\mathcal{P}$ having at least $\sqrt{c/2}$ columns from the v-th block. Let us then remove from $P_{i(v)}$ more columns from the v-th block, so that, for every surviving v, $P_{i(v)}$ always contains exactly $\sqrt{c/2}$ columns from the v-th block.

We then consider the partition $\mathcal{P}'$ containing exactly the parts $P_{i(v)}$ for surviving v, but without any of the columns we have removed thus far, namely, without the high-error columns and without the leftover columns. Let f'_G equal to f_G, but restricted to the surviving columns. Since the error probability of π was $\leq 2n^{-\delta/2}$ on every surviving column, then π will still have error probability $\leq 2n^{-\delta/2}$ on f'_G.

We now remove high-error rows. We first remove each row-block $M_{\{v,w\}}$ such that the error of π on $M_{\{v,w\}}$ is greater than $2n^{-\delta/4}$. Again by Markov's inequality, in doing so we remove $\leq n^{-\delta/4}|E|$ more edges.

Now let E' be the set of surviving edges $\{v, w\}$ such that $i(v) = i(w)$ (i.e. E' contains the low-error edges which violate the coloring constraint). Fix any edge $\{v, w\} \in E'$, and let L be the number of leaves in the 2-round sub-protocol inside part $P_{i(v)} = P_{i(w)}$. If $L \geq \frac{1}{2}s^2 = \Omega(n\ell)$, we then have proven that π has $\Omega(n\ell)$ leaves, and we are done; otherwise, suppose $L < \frac{1}{2}s^2$ leaves. Now notice that, by the gadget property, the surviving columns of the B and C sub-matrices of any such block each have $\frac{2}{3}s$ rows which are pairwise $\frac{1}{10}$-distant in Hamming distance; hence the sub-matrix $[BC]$ within $M_{\{u,v\}}$ containing the surviving columns, must have at least $(\frac{2}{3}s)^2$ rows which are $\frac{1}{20}$-distant in Hamming distance. It then follows from Lemma 29 that the probability of error within $M_{\{u,v\}}$ is $\geq \frac{1/20}{8} = \frac{1}{160}$. And this would happen for every edge $\{u, v\} \in E'$.

It then follows that E' is small. Indeed, if we had $|E'| > n^{-\delta/8} \cdot |E|$, then the total error of π on the surviving sub-matrix would be $\geq \Omega(n^{-\delta/8})$, and since this sub-matrix has $\Omega(1)$ of the total mass of the original matrix, this would contradict π's claimed overall error bound. So we are forced to conclude that $|E'| \leq n^{-\delta/8} \cdot |E|$.

We may then remove all the sub-matrices $M_{\{v,w\}}$ corresponding to edges $\{v, w\} \in E'$. It then follows that the partition $\mathcal{P}'$ is a coloring of the resulting sub-graph G'. Hence $|\mathcal{P}'| \geq \chi(G')$, which by Claim 31 means $|\mathcal{P}'| \geq n^{\frac{\delta}{16}-1}\chi(G)$. Now, within each part P_i of $\mathcal{P}'$, the corresponding A sub-matrix still needs to be solved by an $\mathsf{L}_2^{\mathsf{A}}$-protocol with error $\leq 2n^{-\delta/4}$, which can only be done with ℓ leaves, again by Lemma 29. Hence the total number of leaves is $\Omega(n^{\frac{\delta}{16}-1}\chi(G)\ell)$. ◀

We can then improve upon Theorem 18, and prove that it is NP-hard, under quasipolynomial-time reductions, to distinguish whether a given communication matrix has small deterministic communication complexity, versus large low-error randomized communication complexity. In the next section, we will show that a small improvement of the parameters of the following corollary[4] would be enough to show strong hardness-of-approximation for any number of rounds.

▶ **Corollary 32.** *There exist positive constants γ and δ such that the following holds. There exists a deterministic quasipolynomial-time algorithm that, on input $x \in \{0,1\}^*$, outputs a communication matrix $M \in \{0,1\}^{N\times N}$ and a number $k \in \mathbb{N}$, with $k \le N = |x|^{O(1)}$, such that*

1. *if x is a YES instance of* SAT*, then $\mathsf{L}^{\mathrm{B}}_3(M) \le O(k)$ and $\mathsf{C}^{\mathrm{B}}_3(M) \le \log k + O(1)$, and*
2. *if x is a NO instance of* SAT*, then $\mathsf{L}^{\mathrm{B}}_{3,N^{-\delta}}(M) \ge \Omega(N^\gamma \cdot k)$ and $\mathsf{C}^{\mathrm{B}}_{3,N^{-\delta}}(M) \ge \log k + \gamma \cdot \log N - O(1)$.*

Proof. We will choose $\delta_3 = \delta > 0$ to be a sufficiently small constant. We combine the two reductions of Theorems 9 and 30, which we invoke with parameter $\varepsilon = \delta/64$. Fix any input x and let G be an n-vertex graph that is produced by the reduction of Theorem 9 on input x. We have $n = |x|^{O(1)}$ since the reduction of Theorem 9 is polytime. Let $M \in \{0,1\}^{a\times b}$ be the communication matrix of f_G, and $\ell \in \mathbb{N}$, be given by the reduction of Theorem 17 on input G, and set $N = \max(a,b) = O(n^{27})$, $k = n^\varepsilon \cdot \ell$. We may assume that $a = b$ without loss of generality, since otherwise we may pad M with all-0 rows or all-0 columns, and this changes the leaf complexity by at most a factor of 2, and the communication complexity by at most 1 bit, while leaving the error parameter intact.

We verify that the inequalities are satisfied for M: If $x \in L$, then we have $\mathsf{L}^{\mathrm{B}}_3(M) = O(\chi(G)\cdot \ell) = O(k)$. If $x \notin L$, then we have

$$\begin{aligned}
\mathsf{L}^{\mathrm{B}}_{3,n^{-\delta}}(M) &= \Omega(n^{\frac{\delta}{16}-1}\chi(G)\cdot \ell)\\
&= \Omega(n^{\frac{\delta}{16}-\varepsilon}\cdot \ell)\\
&= \Omega(n^{\frac{\delta}{16}-2\varepsilon}\cdot k)\\
&= \Omega(n^{\frac{\delta}{32}}\cdot k)\\
&= \Omega(N^{\frac{\delta}{32\times 27}}\cdot k)
\end{aligned}$$

Similar inequalities hold for $\mathsf{C}^{\mathrm{B}}_3(M)$. So we set $\gamma = \frac{\delta}{32\times 27}$. ◀

7 From 3-rounds to multiple rounds using round elimination

We would now like to prove that constant-round communication complexity is NP-hard, for any number of rounds. However, the result we proved in Corollary 32 is not enough. We conjecture that the parameters in that result can be improved, as follows

▶ **Conjecture 33.** *For any constant $C \ge 1$, there exist positive constants $\gamma, \delta \in (0,1]$ with $\gamma \ge C\cdot\delta$ such that the following holds. There exists a deterministic quasipolynomial-time algorithm that, on input $x \in \{0,1\}^*$, outputs a communication matrix $M \in \{0,1\}^{N\times N}$ and a number $k \in \mathbb{N}$, with $k \le N = |x|^{O(1)}$, such that*

1. *if x is a YES instance of* SAT*, then $\mathsf{L}^{\mathrm{B}}_3(M) \le O(k)$ and $\mathsf{C}^{\mathrm{B}}_3(M) \le \log k + O(1)$, and*
2. *if x is a NO instance of* SAT*, then $\mathsf{L}^{\mathrm{B}}_{3,N^{-\delta}}(M) \ge \Omega(N^\gamma \cdot k)$ and $\mathsf{C}^{\mathrm{B}}_{3,N^{-\delta}}(M) \ge \log k + \gamma \cdot \log N - O(1)$.*

[4] As we will see, it would be enough if γ could be made arbitrarily larger than δ.

Let us devise notation that will help us better understand the difference. We may now define the following problems:

▶ **Definition 34.** *In the problem* $\mathsf{MPL}^{\mathrm{A}}(d, \varepsilon, \phi, N)$*, defined for each natural number* $d \geq 3$*, all* $\varepsilon \in [0,1]$*,* $\phi \geq 1$*, and* $N \in \mathbb{N}$*, we are given as input an* $N \times N$ *Boolean matrix* M*, and a natural number* $1 \leq k \leq N$*, with the promise that*

- *either* $\mathsf{L}^{\mathrm{A}}_{d}(M) \leq k$*,*
- *or* $\mathsf{L}^{\mathrm{A}}_{d,\varepsilon}(M) \geq \phi \cdot k$

and we wish to decide which is the case. We define $\mathsf{MPL}^{\mathrm{B}}$ *in the same way.*

In the problem $\mathsf{MPC}^{\mathrm{A}}(d, \varepsilon, \phi, K, N)$*, defined for each natural number* $d \geq 3$*, all* $\varepsilon, \phi \in [0,1]$*, and all* $K, N \in \mathbb{N}$*, we are given as input an* $N \times N$ *Boolean matrix* M*, and a natural number* $1 \leq k \leq K$*, with the promise that*

- *either* $\mathsf{L}^{\mathrm{A}}_{d}(M) \leq k$*,*
- *or* $\mathsf{L}^{\mathrm{A}}_{d,\varepsilon}(M) \geq k + \phi \cdot K$

and we wish to decide which is the case. We define $\mathsf{MPC}^{\mathrm{B}}$ *analogously.*

Then Corollary 32 and Conjecture 33 tell us that these approximation problems are NP-hard, for different parameters. In this notation we may restate Corollary 32 and Conjecture 33 as follows:

- (Corollary 32) There exist positive constants γ and δ such that $\mathsf{MPL}^{\mathrm{A}}(3, N^{-\delta}, N^{\gamma}, N)$ and $\mathsf{MPC}^{\mathrm{A}}(3, N^{-\delta}, \gamma, \log N, N)$ are NP-hard under deterministic quasipolynomial-time many-one reductions.
- (Conjecture 33) For any constant $C \geq 1$, there exist positive constants $\gamma, \delta \in (0,1]$ with $\gamma \geq C \cdot \delta$ such that, $\mathsf{MPL}^{\mathrm{A}}(3, N^{-\delta}, N^{\gamma}, N)$ and $\mathsf{MPC}^{\mathrm{A}}(3, N^{-\delta}, \gamma, \log N, N)$ are NP-hard under deterministic quasipolynomial-time many-one reductions.

In the rest of this section, we will use Conjecture 33 and the round elimination lemma to prove that quasipolynomial-time algorithms for computing constant-round communication complexity would place all of NP in subexponential time.

We begin by recalling the round-elimination lemma, which was originally proven by Miltersen, Nisan, Safra, and Wigderson [46] and later improved and simplified by Sen and Venkatesh [62], using an information-theoretic argument. Sen and Venkatesh's proof is already information-theoretic in flavor, but it can be made significantly shorter by using a nowadays-standard combination of the chain rule and Pinsker's inequality (see [15, 58]). We include this shortened proof here, on the one hand so that we can confirm that it works for leaf complexity, and not just communication complexity, and on the other hand so we can extract the exact parameters.

▶ **Theorem 35** (Round Elimination Lemma [46, 62])**.** *Let* $3 \leq d \in \mathbb{N}$ *and let* $\alpha > 0$*. Given a Boolean function* $f : [a] \times [b] \to \{0,1\}$ *and a parameter* $\beta > 0$*, we may construct a Boolean function* $F : [a]^m \times ([m] \times [b]) \to \{0,1\}$*, with* $m = \frac{1}{4\beta^2}(\lceil \log \min(a,b) \rceil + 1)$*, such that*

$$\mathsf{L}^{\mathrm{A}}_{d+1}(F) \leq m \cdot \mathsf{L}^{\mathrm{B}}_{d}(f), \qquad \mathsf{L}^{\mathrm{A}}_{d+1,\alpha}(F) \leq m \cdot \mathsf{L}^{\mathrm{B}}_{d,\alpha}(f), \qquad \mathsf{L}^{\mathrm{B}}_{d,\alpha}(f) \leq \mathsf{L}^{\mathrm{A}}_{d+1,\alpha-\beta}(F),$$

and also

$$\mathsf{C}^{\mathrm{A}}_{d+1}(F) \leq \mathsf{C}^{\mathrm{B}}_{d}(f) + \lceil \log m \rceil \qquad \mathsf{C}^{\mathrm{A}}_{d+1,\alpha}(F) \leq \mathsf{C}^{\mathrm{B}}_{d,\alpha}(f) + \lceil \log m \rceil \qquad \mathsf{C}^{\mathrm{B}}_{d,\alpha}(f) \leq \mathsf{C}^{\mathrm{A}}_{d+1,\alpha-\beta}(F).$$

Proof. We define $F(x_1, \ldots, x_m; i, y) = f(x_i; y)$. Meaning, Alice is given m Alice-side inputs $x_1, \ldots, x_m \in [a]$ for f, and Bob is given an index $i \in [m]$ and a Bob-side input $y \in [b]$ for f, and they wish to compute $f(x_i; y)$.

The upper-bounds on $\mathsf{L}^{\mathrm{A}}_{d+1}$ and $\mathsf{C}^{\mathrm{A}}_{d+1}$ are simple to see. Indeed, a $d+1$ round protocol where Alice begins to speak may have Alice send nothing in her first round, after which Bob sends i to Alice and then the players may compute $f(x; y_i)$ by executing a d-round protocol for f. This works whether or not the protocol for f is randomized.

Now to prove the lower-bounds on $\mathsf{L}^{\mathrm{A}}_{d+1}$ and $\mathsf{C}^{\mathrm{A}}_{d+1}$. To prove the lower-bound on the leaf complexity, we may assume that $\mathsf{L}^{\mathrm{A}}_{d+1,\alpha-\beta}(F) \leq 2\min(a,b)$, and to prove the lower-bound on the communication complexity, we may assume that $\mathsf{C}^{\mathrm{A}}_{d+1,\alpha-\beta}(F) \leq \lceil \log \min(a,b) \rceil + 1$, since otherwise the respective inequalities are trivial. Let μ be the hard distribution for f. Construct a distribution μ' for F by choosing $i \in [m]$ uniformly at random, sampling (x_i, y) according to μ, and then sampling each x_j with $j \neq i$ from the x-marginal of μ. Now suppose we are given a deterministic Alice-first $(d+1)$-round protocol π' for F with $\mathsf{L}(\pi') \leq 2\min(a,b)$ leaves and communication complexity $\mathsf{C}(\pi') \leq \lceil \log \min(a,b) \rceil + 1$, and which makes $\alpha - \beta$ error (or less) when the input is sampled according to μ', and let us construct a deterministic Bob-first d-round protocol π for f having $\mathsf{L}(\pi) \leq \mathsf{L}(\pi')$ leaves and communication complexity $\mathsf{C}(\pi) \leq \mathsf{C}(\pi')$, which makes α error when the input is sampled according to μ.

Let $(\mathbf{x}_1, \ldots, \mathbf{x}_m; \mathbf{i}, \mathbf{y})$ denote random variables sampled according to the distribution μ', and let $\mathbf{t} = \mathbf{t}(\mathbf{x}_1, \ldots, \mathbf{x}_m)$ be the message sent in the first round of π'. Let T denote either $\log \mathsf{L}(\pi')$ or $\mathsf{C}(\pi')$, so that $T \leq \lceil \log \min(a,b) \rceil + 1$. We then have, by the chain rule,

$$T \geq \mathsf{H}(\mathbf{t}) \geq \mathsf{I}(\mathbf{x}_1, \ldots, \mathbf{x}_m : \mathbf{t}) = \sum_{i=1}^{m} \mathsf{I}(\mathbf{x}_i : \mathbf{t} \mid \mathbf{x}_{<i}) = m \cdot \mathsf{I}(\mathbf{x}_{\mathbf{i}} : \mathbf{t} \mid \mathbf{i}, \mathbf{x}_{<\mathbf{i}})$$

Hence there exists a setting $\mathbf{i} = i$ such that

$$\mathsf{I}(\mathbf{x}_{\mathbf{i}} : \mathbf{t} \mid \mathbf{i} = i, \mathbf{x}_{<\mathbf{i}}) \leq \frac{1}{m} T \leq 2\beta^2.$$

Let us then fix some setting $\mathbf{x}_{<i} = x_{<i}$ which attains the above bound, so that

$$\mathsf{I}(\mathbf{x}_{\mathbf{i}} : \mathbf{t} \mid \mathbf{i} = i, \mathbf{x}_{<\mathbf{i}} = x_{<i}) \leq 2\beta^2$$

By Pinsker's inequality, this implies that the average, over choices t for $\mathbf{t}$, statistical distance between $\mathbf{x}_i$ and $\mathbf{x}_i$ conditioned on $\mathbf{t} = t$, is at most β. On the other hand, the average error (of π on μ') over choices of t for $\mathbf{t}$ is at most $\alpha - \beta$. By linearity of expectation, the average sum of the error plus statistical distance is at most α. Let us then fix a choice t for $\mathbf{t}$ where this sum is at most α.

We may then consider the Bob-first d-round protocol $\tilde{\pi}$ that executes the last d-rounds of π, for the case when the first message of π is t, where $\mathbf{x}_{<i}$ has been fixed to $x_{<i}$, and each coordinate of $\mathbf{x}_{>i}$ is chosen at random from the x-marginal of μ. Such a protocol will incur a total error $\leq \alpha$, and has fewer leaves ans smaller communication complexity than π'. ◀

The round-elimination lemma can be used to reduce the computation of complexity on d rounds to the computation of complexity on $d+1$ rounds, as follows.

▶ **Corollary 36.** *We may reduce* $\mathsf{MPL}^{\mathrm{A}}(d, \varepsilon, \phi, N)$ *to* $\mathsf{MPL}^{\mathrm{A}}(d+1, \frac{\varepsilon}{2}, \frac{\phi}{m}, N^m)$*, and we may reduce* $\mathsf{MPC}^{\mathrm{A}}(d, \varepsilon, \phi, K, N)$ *to* $\mathsf{MPC}^{\mathrm{A}}(d+1, \frac{\varepsilon}{2}, \phi - 2\frac{\log m}{K}, K + \log m, N^m)$*, where* $m = 32\varepsilon^{-2} \log N$*, by a many-one reduction computable in time* $N^{O(m)}$*.*

Proof. The reduction is given an $N \times N$ communication matrix M, which corresponds to a Boolean function $f : [N] \times [N] \to \{0,1\}$, and a number $k \leq N$. Let F given by Theorem 35, with parameters $m = 32\varepsilon^{-2} \log N$, $\alpha = \varepsilon$ and $\beta = \frac{\varepsilon}{2}$. Then the output is a matrix $M' \in \{0,1\}^{N' \times N'}$ where $N' \leq N^m$, obtained from the communication matrix of F padded with extra 0-columns to make it into a square matrix.

Then if $\mathsf{L}^{\mathrm{A}}_{d}(M) \leq k$, we will also have $\mathsf{L}^{\mathrm{A}}_{d+1}(M') \leq mk$. On the other hand, if $\mathsf{L}^{\mathrm{A}}_{d,\alpha}(M) = \mathsf{L}^{\mathrm{A}}_{d,\varepsilon}(M) \geq \phi \cdot k$, then $\mathsf{L}^{\mathrm{A}}_{d+1,\frac{\varepsilon}{2}}(M') = \mathsf{L}^{\mathrm{A}}_{d+1,\alpha-\beta} \geq \mathsf{L}^{\mathrm{A}}_{d,\varepsilon}(M) \geq \phi \cdot k = \frac{\phi}{m} \cdot mk$.

Furthermore if $\mathsf{C}^{\mathrm{A}}_{d}(M) \leq k$, we will also have $\mathsf{C}^{\mathrm{A}}_{d+1}(M') \leq k + \log m$. On the other hand, if $\mathsf{C}^{\mathrm{A}}_{d,\alpha}(M) = \mathsf{C}^{\mathrm{A}}_{d,\varepsilon}(M) \geq k + \phi \cdot K$, then $\mathsf{C}^{\mathrm{A}}_{d+1,\frac{\varepsilon}{2}}(M') = \mathsf{C}^{\mathrm{A}}_{d+1,\alpha-\beta} \geq \mathsf{C}^{\mathrm{A}}_{d,\varepsilon}(M) \geq k + \phi \cdot K$, and

$$\begin{aligned} k + \phi \cdot K &= k + \log m + \left(\phi - \frac{(1+\phi)\log m}{K + \log m}\right) \cdot (K + \log m) \\ &\geq k + \log m + \left(\phi - 2\frac{\log m}{K}\right) \cdot (K + \log m). \end{aligned}$$ ◀

We can now show that if communication complexity C^{A} (and not just leaf complexity L^{A}) can be computed in quasipolynomial time and Conjecture 33 holds, then all of NP can be solved in subexponential time, i.e., time $2^{n^{\varepsilon}}$, for any choice $\varepsilon > 0$. A similar result can be proven for leaf complexity, with better hardness-of-approximation than what is obtained in Section 5, but we will omit the proof here, because it is very similar, and we already have the results of Section 5. If the error parameter in Conjecture 33 can be made into a constant, instead of $N^{-\delta}$, then the same proof will give us quasipolynomial NP-hardness instead of subexponential. We also omit this proof.

▶ **Theorem 37.** *If Conjecture 33 holds and* C^{A} *can be computed in quasipolynomial time, then all of* NP *can be computed in subexponential time.*

Proof. Conjecture 33 says that a SAT instance of size n reduces to $\mathsf{MPC}^{\mathrm{A}}(3, N^{-\delta}, \gamma, \log N, N)$ with $N = n^{O(1)}$, for any choice of γ, δ, where δ can be chosen to be arbitrarily small, and γ can be chosen to be as many times higher than δ as needed. We may now apply Corollary 36 repeatedly. We are only aiming for rough parameters, and so for simplicity, in the first application, we replace $\log N$ factors with N^{δ}, and in all applications, we replace the 32 factor with N^{δ}, as well. We can do this because $N^{\delta} \gg \log N$, and $\mathsf{MPC}^{\mathrm{A}}(d, \varepsilon, \phi, K, N)$ reduces to $\mathsf{MPC}^{\mathrm{A}}(d, \varepsilon, \phi', K, N')$ whenever $\phi' \leq \phi$ and $N' \geq N$.

We then find that $\mathsf{MPC}^{\mathrm{A}}(3, N^{-\delta}, \gamma, \log N, N)$ reduces to

$$\mathsf{MPC}^{\mathrm{A}}(4, \frac{1}{2}N^{-\delta}, \gamma - 8\delta, (1 + 4\delta)\log N, 2^{N^{5\delta}}),$$

which reduces to

$$\mathsf{MPC}^{\mathrm{A}}(5, 2^{-2} \cdot N^{-\delta}, \gamma - 16\delta, 2^{N^{13\delta}}),$$

etc, which reduces to

$$\mathsf{MPC}^{\mathrm{A}}(d, C_1 \cdot N^{-\delta}, \gamma - C_2\delta, (1 + C_3\delta)\log N, 2^{N^{C_4\delta}})$$

for some positive constants C_1, C_2, C_3, C_4 that depend only on df. This problem in turn reduces to computing $\mathsf{C}^{\mathrm{A}}(f)$ exactly, on instances $f : [N'] \times [N'] \to \{0, 1\}$, for $N' = 2^{N^{C_4\delta}}$. Now suppose we could compute $\mathsf{C}^{\mathrm{A}}(f)$ exactly in time quasipolynomial in $N' = 2^{N^{C_4\delta}}$, i.e., in time $2^{N^{O(C_4\delta)}}$. Then by choosing δ to be sufficiently small, given that $N = n^{O(1)}$, we could then solve SAT in time $2^{n^{\varepsilon}}$, for any $\varepsilon > 0$ of our choosing. ◀

References

1 Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *ACM Transactions on Computation Theory*, 1(1):2, 2009.

2 Miklós Ajtai. Σ^1_1-formulae on finite structures. *Annals of pure and applied logic*, 24(1):1–48, 1983.

3 Miklós Ajtai and Michael Ben-Or. A Theorem on Probabilistic Constant Depth Computations. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 471–474, 1984.

4 Eric Allender. The new complexity landscape around circuit minimization. In Alberto Leporati, Carlos Martín-Vide, Dana Shapira, and Claudio Zandron, editors, *Language and Automata Theory and Applications* (LATA), volume 12038 of *Lecture Notes in Computer Science*, pages 3–16. Springer, 2020.

5 Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. In *International Symposium on Mathematical Foundations of Computer Science* (MFCS), pages 25–32, 2014.

6 Eric Allender, Joshua A. Grochow, Dieter van Melkebeek, Cristopher Moore, and Andrew Morgan. Minimum circuit size, graph isomorphism, and related problems. *SIAM Journal on Computing*, 47(4):1339–1372, 2018.

7 Eric Allender, Lisa Hellerstein, Paul McCabe, Toniann Pitassi, and Michael E. Saks. Minimizing disjunctive normal form formulas and AC^0 circuits given a truth table. *SIAM Journal on Computing*, 38(1):63–84, 2008.

8 Eric Allender and Shuichi Hirahara. New insights on the (non-)hardness of circuit minimization and related problems. In *International Symposium on Mathematical Foundations of Computer Science* (MFCS), pages 54:1–54:14, 2017.

9 Eric Allender, Dhiraj Holden, and Valentine Kabanets. The minimum oracle circuit size problem. *Computational Complexity*, 26(2):469–496, 2017.

10 Eric Allender, Rahul Ilango, and Neekon Vafa. The non-hardness of approximating circuit size. In *International Computer Science Symposium in Russia* (CSR), pages 13–24, 2019.

11 Eric Allender and Michal Kouckỳ. Amplifying lower bounds by means of self-reducibility. *Journal of the ACM*, 57(3):1–36, 2010.

12 Eric Allender, Michal Kouckỳ, Detlef Ronneburger, and Sambuddha Roy. The pervasive reach of resource-bounded Kolmogorov complexity in computational complexity theory. *Journal of Computer and System Sciences*, 77(1):14–40, 2011.

13 Noga Alon and Joel H Spencer. *The probabilistic method.* John Wiley & Sons, 2016.

14 Theodore Baker, John Gill, and Robert Solovay. Relativizations of the p=?np question. *SIAM Journal on computing*, 4(4):431–442, 1975.

15 Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. *SIAM Journal on Computing*, 42(3):1327–1363, 2013.

16 Avrim L. Blum and Ronald L. Rivest. Training a 3-node neural network is np-complete. *Neural Networks*, 5(1):117–127, 1992.

17 Joan Boyar, Philip Matthews, and René Peralta. On the shortest linear straight-line program for computing linear forms. In *International Symposium on Mathematical Foundations of Computer Science* (MFCS), pages 168–179, 2008.

18 Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning algorithms from natural proofs. In *Conference on Computational Complexity* (CCC), pages 10:1–10:24, 2016.

19 Sebastian Lukas Arne Czort. The complexity of minimizing disjunctive normal form formulas. Master's thesis, University of Aarhus, 1999.

20 Uriel Feige and Joe Kilian. Zero knowledge and the chromatic number. *Journal of Computer and System Sciences*, 57(2):187–199, 1998.

21 Vitaly Feldman. Hardness of approximate two-level logic minimization and PAC learning with membership queries. In *Symposium on Theory of Computing (STOC)*, pages 363–372, 2006.

22 Alexander Golovnev, Rahul Ilango, Russell Impagliazzo, Valentine Kabanets, Antonina Kolokolova, and Avishay Tal. $ac^0[p]$ lower bounds against mcsp via the coin problem. In *Colloquium on Automata, Languages, and Programming* (ICALP), volume 132, page 66, 2019.

23 Thomas R. Hancock, Tao Jiang, Ming Li, and John Tromp. Lower bounds on learning decision lists and trees. *Information and Computation*, 126(2):114–122, 1996.

24 Johan Hastad. Clique is hard to approximate within $n^{1-\varepsilon}$. In *Symposium on Foundations of Computer Science* (FOCS), pages 627–636, 1996.

25 Shuichi Hirahara, Igor Carboni Oliveira, and Rahul Santhanam. NP-hardness of minimum circuit size problem for OR-AND-MOD circuits. In *Computational Complexity Conference* (CCC), pages 5:1–5:31, 2018.

26 Shuichi Hirahara and Osamu Watanabe. Limits of minimum circuit size problem as oracle. In *Conference on Computational Complexity* (CCC), pages 18:1–18:20, 2016.

27 John M. Hitchcock and Aduri Pavan. On the NP-completeness of the minimum circuit size problem. In *Foundation of Software Technology and Theoretical Computer Science* (FSTTCS), pages 236–245, 2015.

28 Rahul Ilango. Approaching MCSP from above and below: Hardness for a conditional variant and $ac^0[p]$. In Thomas Vidick, editor, *Innovations in Theoretical Computer Science Conference* (ITCS), volume 151 of *LIPIcs*, pages 34:1–34:26. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

29 Rahul Ilango. Constant depth formula and partial function versions of MCSP are hard. In *Proceedings of the Symposium on Foundations of Computer Science* (FOCS), pages 424–433. IEEE, 2020.

30 Rahul Ilango, Bruno Loff, and Igor Carboni Oliveira. Np-hardness of circuit minimization for multi-output functions. In Shubhangi Saraf, editor, *Computational Complexity Conference* (CCC), volume 169 of *LIPIcs*, pages 22:1–22:36. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

31 Russell Impagliazzo, Valentine Kabanets, and Ilya Volkovich. The Power of Natural Properties as Oracles. In *Computational Complexity Conference* (CCC), volume 102, pages 7:1–7:20, 2018.

32 J. Stephen Judd. Learning in networks is hard. In *International Conference on Neural Networks* (ICNN), volume 2, pages 685–692, 1987.

33 Valentine Kabanets and Jin-yi Cai. Circuit minimization problem. In *Symposium on Theory of Computing* (STOC), pages 73–79, 2000.

34 Mauricio Karchmer, Eyal Kushilevitz, and Noam Nisan. Fractional covers and communication complexity. *SIAM Journal on Discrete Mathematics*, 8(1):76–92, 1995.

35 Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, page 539–550. Association for Computing Machinery, 1988.

36 Subhash Khot and Rishi Saket. Hardness of minimizing and learning DNF expressions. In *Symposium on Foundations of Computer Science* (FOCS), pages 231–240, 2008.

37 Jan Krajícek. *Forcing with Random Variables and Proof Complexity*. Cambridge University Press, 2011.

38 Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

39 Eyal Kushilevitz and Enav Weinreb. On the complexity of communication complexity. In *Symposium on Theory of Computing* (STOC), pages 465–474, 2009.

40 Yanyi Liu and Rafael Pass. On one-way functions and kolmogorov complexity. *arXiv preprint*, 2020. `arXiv:2009.11514`.

41 L. Lovasz and M. Saks. Lattices, mobius functions and communications complexity. In *Symposium on Foundations of Computer Science* (FOCS), SFCS '88, page 81–90, USA, 1988. IEEE Computer Society.

42 Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.

43 William J. Masek. Some NP-complete set covering problems. Unpublished Manuscript, 1979.

44 Dylan M. McKay, Cody D. Murray, and R. Ryan Williams. Weak lower bounds on resource-bounded compression imply strong separations of complexity classes. In *Symposium on Theory of Computing* (STOC), STOC 2019, page 1215–1225, New York, NY, USA, 2019. Association for Computing Machinery.

45 Dylan M. McKay, Cody D. Murray, and R. Ryan Williams. Weak lower bounds on resource-bounded compression imply strong separations of complexity classes. In *Symposium on Theory of Computing* (STOC), 2019.

46 Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. *Journal of Computer and System Sciences*, 57(1):37–49, 1998.

47 Moritz Müller and Ján Pich. Feasibly constructive proofs of succinct weak circuit lower bounds. *Annals of Pure and Applied Logic*, 171(2), 2020.

48 Cody D. Murray and Richard Ryan Williams. On the (non) NP-hardness of computing circuit complexity. In *Conference on Computational Complexity* (CCC), pages 365–380, 2015.

49 Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM journal on computing*, 22(4):838–856, 1993.

50 Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *Journal of the ACM*, 51(2):231–262, 2004.

51 Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and Near-Optimal Derandomization. In *Symposium on Foundations of Computer Science* (FOCS), pages 182–191, 1995.

52 Noam Nisan. Pseudorandom bits for constant depth circuits. *Combinatorica*, 11(1):63–70, 1991.

53 Igor Carboni Oliveira, Ján Pich, and Rahul Santhanam. Hardness magnification near state-of-the-art lower bounds. In *Conference on Computational Complexity* (CCC), 2019.

54 Igor Carboni Oliveira and Rahul Santhanam. Conspiracies between learning algorithms, circuit lower bounds, and pseudorandomness. In *Computational Complexity Conference* (CCC), pages 18:1–18:49, 2017.

55 Igor Carboni Oliveira and Rahul Santhanam. Hardness magnification for natural problems. In *Symposium on Foundations of Computer Science* (FOCS), pages 65–76, 2018.

56 Denis Pankratov. Direct sum questions in classical communication complexity. *Master's thesis, University of Chicago*, 2012.

57 Leonard Pitt and Leslie G. Valiant. Computational limitations on learning from examples. *Journal of the ACM*, 35(4):965–984, 1988.

58 Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020.

59 Alexander A Razborov. On submodular complexity measures. *Boolean Function Complexity,(M. Paterson, Ed.)*, pages 76–83, 1992.

60 Alexander A Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.

61 Alexander A. Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.

62 Pranab Sen and Srinivasan Venkatesh. Lower bounds for predecessor searching in the cell probe model. *Journal of Computer and System Sciences*, 74(3):364–385, 2008.

63 Boris A Trakhtenbrot. A survey of Russian approaches to perebor (brute-force search) algorithms. *Annals of the History of Computing*, 6(4):384–400, 1984.

64 Christopher Umans, Tiziano Villa, and Alberto L. Sangiovanni-Vincentelli. Complexity of two-level logic minimization. *IEEE Transactions on CAD of Integrated Circuits and Systems*, 25(7):1230–1246, 2006.

65 David Zuckerman. Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number. *Theory of Computing*, 3(1):103–128, 2007.