Electronic Colloquium on Computational Complexity, Report No. 170 (2017)

Simulation Beats Richness: New Data-Structure Lower Bounds

Arkadev Chattopadhyay*
TIFR, Mumbai

Michal Koucký†
Charles University, Prague

Bruno Loff‡
INESC-Tec and University of Porto

Sagnik Mukhopadhyay§
KTH Royal Institute of Technology

Abstract

We develop a technique for proving lower bounds in the setting of asymmetric communication, a model that was introduced in the famous works of Miltersen (STOC'94) and Miltersen, Nisan, Safra and Wigderson (STOC'95). At the core of our technique is a novel simulation theorem: Alice gets a $p \times n$ matrix x over $\mathbb{F}_2$ and Bob gets a vector $y \in \mathbb{F}_2^n$. Alice and Bob need to evaluate $f(x \cdot y)$ for a Boolean function $f : \{0,1\}^p \to \{0,1\}$. Our simulation theorems show that a deterministic/randomized communication protocol exists for this problem, with cost $C \cdot n$ for Alice and C for Bob, if and only if there exists a deterministic/randomized *parity decision tree* of cost $\Theta(C)$ for evaluating f.

As applications of this technique, we obtain the following results:

1. The first strong lower-bounds against randomized data-structure schemes for the Vector-Matrix-Vector product problem over $\mathbb{F}_2$. Moreover, our method yields strong lower bounds even when the data-structure scheme has tiny advantage over random guessing.
2. The first lower bounds against randomized data-structures schemes for two natural Boolean variants of Orthogonal Vector Counting.
3. We construct an asymmetric communication problem and obtain a deterministic lower-bound for it which is provably better than any lower-bound that may be obtained by the classical Richness Method of Miltersen et al. [MNSW98]. This seems to be the first known limitation of the Richness Method in the context of proving deterministic lower bounds.

*arkadev.c@tifr.res.in — partially funded by a Ramanujan Fellowship of the DST, India.

†koucky@iuuk.mff.cuni.cz— The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013)/ERC Grant Agreement n. 616787. The author was partially supported by the Center of Excellence CE-ITI (P202/12/G061 of GA ČR).

‡bruno.loff@gmail.com — Funded by FCT posdoctoral grant number SFRH/BPD/116010/2016.

§sagnik@kth.se — A lot of the work done while the author was graduate student of TIFR and a recipient of a TCS fellowship.

ISSN 1433-8092

Contents

1 Introduction

A central question in theoretical computer science is proving lower bounds on the time needed to solve various algorithmic problems. For general computation this is extremely difficult; indeed, over the past many decades there has been only limited progress in this area despite great effort. One of the main available techniques to prove such lower bounds is the analysis of the flow of information during computation. The area of communication complexity is devoted entirely to the analysis of this information flow.

Data structure problems are computational problems having a well structured form, where information bottlenecks can often be found. Communication complexity is the key technique to prove them. In a static data structure problem we have a domain $\mathcal{D}$ of possible data, a domain $\mathcal{Q}$ of possible queries and a function $f : \mathcal{D} \times \mathcal{Q} \to \mathcal{A}$ where $f(x; y)$ represents the answer to query y on data x. The goal is to store the data x in memory, using space as efficiently as possible, so that given a query y we can evaluate $f(x; y)$ quickly.[1] A major theme of research is to understand the space-query tradeoffs inherent to natural problems.

This paper explores this theme in data structures with problems related to matrix-vector multiplication. In the vector-matrix-vector problem $\mathsf{VMV}_{n\times n}$, the data are matrices $x \in \mathbb{F}^{n^2}$ over some field $\mathbb{F}$, queries are pairs of vectors $(q, y) \in \mathbb{F}^n \times \mathbb{F}^n$, and the solicited answers are $f(x; (q, y)) = q \cdot x \cdot y$. In the orthogonal vector counting problem $\mathsf{OVC}_{n\times n}$, the data is also a matrix $x \in \mathbb{F}^{n^2}$, the query is a single vector $y \in \mathbb{F}^n$ and $f(x; y)$ counts the number of zeros in $x \cdot y$, i.e., the number of rows of x which are orthogonal to y; we will actually consider two different variants of OVC which have a 1-bit output. The mod-3 orthogonal vector counting $\mathsf{OVC}^3_{n\times n}$ is a variant of $\mathsf{OVC}_{n\times n}$ where $f(x; y) = 1$ if the number of rows of x which are orthogonal to y is a multiple of 3, and $f(x; y) = 0$ otherwise. The orthogonal gap-majority problem $\mathsf{OGMaj}_{n\times n}$ is a promise variant of $\mathsf{OVC}_{n\times n}$, where we have $f(x; y) = 1$ if at least $\frac{n}{2} + \sqrt{n}$ of the rows of x are orthogonal to y, and $f(x; y) = 0$ if no more than $\frac{n}{2} - \sqrt{n}$ of the rows of x are orthogonal to y, with the promise that we are in one of the two cases.

We are interested in the complexity of these data-structure problems in Yao's *cell-probe model* [Yao79]. In this model the data is represented in a memory consisting of s cells, each cell storing w bits. We do not charge for the preprocessing time to create the data structure in memory for given x, but we charge for the time to answer a query y. The cost of the query is the number of memory cells we have to read (probe) in order to answer the query. This model is one of the most general data structure models; in particular, any lower bound on the number of probes to answer a query immediately translates into a lower bound on the time to answer a query in models such as the word-RAM.

The problems we study are closely related to previous work on matrix-vector product. Henzinger et al. [HKNS15], and Larsen and Williams [LW17] study the matrix-vector product and the vector-matrix-vector product over the *Boolean semiring*, in its relation to fine-grained complexity and conditional lower bounds. In particular, Henzinger et al. conjecture that there are no *truly subcubic* algorithms to solve the online version of matrix-vector multiplication (OMV). Assuming this conjecture, they are able to establish tight lower bounds for over a dozen different dynamic problems, establishing the central importance that OMV enjoys in this area. Indeed, unconditional lower bounds for some versions of matrix-vector multiplication have been recently established. Frandsen et al. [FHM01] study the matrix-vector multiplication over finite fields, and give a lower bound $\Omega(\min\{\frac{n \log |\mathbb{F}|}{\log s}, n^2\})$ on the number of cell-probes for deterministic data structures, where $|\mathbb{F}|$ is the field size. Clifford, Grønlund and Larsen [CGL15] improved this to $\Omega(\min\{\frac{n \log |\mathbb{F}|}{\log(s/n)}, n^2\})$ in

[1] In dynamic data structures we also allow certain updates to the data x.

the randomized setting even with error $1 - |\mathbb{F}|^{n/4}$ for fields of size $|\mathbb{F}| = n^{\Omega(1)}$ and $w = \Theta(\log |\mathbb{F}|)$. Interestingly, while there exist several hardness results for different versions of matrix-vector multiplication problem, there are no strong randomized hardness result for the VMV problem. The difficulty might be in the fact that the output of VMV is merely 1 bit. This obstacle of proving lower bounds for decision problems is not isolated to static data-structure problems. In the setting of *dynamic* data-structure problems, a breakthrough result of Larsen [Lar12] established the first super-logarithmic dynamic lower bounds for a *non-Boolean* problem more than five years ago, but the analogous bound for a Boolean valued decision problem had to wait until a very recent breakthrough of Larsen, Weinstein and Yu [LWY17].

Indeed it is not clear that the hardness of matrix-vector product (MVP) carries over to VMV unabated. For instance, while Larsen and Williams [LW17] give a surprising data structure for VMV over the Boolean semiring which uses only $O(n^{3/2}/\sqrt{w})$ cell probes to answer a query, their upper-bounds for MVP in the same setting require a larger number $O(n^{7/4}/\sqrt{w})$ of cell probes.

The problem of counting orthogonal vectors has been widely studied in the context of fine-grained complexity [CW16, WY14, DL17], although in that setting the dimension of the input vectors is much smaller than the number of vectors, and these two are comparable in our setting.

1.1 Data-structure lower-bounds

We study the VMV, OVC and OGMaj problems over the field $\mathbb{F}_2 = \mathrm{GF}[2]$. We establish the following new lower-bounds against randomized data-structure schemes:

Theorem I. *There exists a real constant $\varepsilon > 0$ such that:*

(a) Any randomized data-structure scheme for $\mathsf{VMV}_{n\times n}$ that uses s cells, each storing $w \le n$ bits, must either make $t \ge \frac{\varepsilon n}{\log \frac{sw}{n}}$ probes, or have success probability $\rho \le \frac{1}{2} + 2^{-\varepsilon n}$.

(b) Any randomized data-structure scheme for $\mathsf{OVC}^3_{n\times n}$ that uses s cells, each storing $w \le n$ bits, must either make $t \ge \frac{\varepsilon n}{\log \frac{sw}{n}}$ probes, or have success probability $\rho \le \frac{2}{3} + 2^{-\varepsilon n}$. [2]

(c) Any randomized data-structure scheme for $\mathsf{OGMaj}_{n\times n}$ that uses s cells, each storing $w \le n$ bits, must either make $t \ge \frac{\varepsilon n}{\log \frac{sw}{n}}$ probes, or have success probability $\rho \le 1 - \varepsilon$.

The above lower bounds are optimal when the cell size $w = n$, as each problem above has a deterministic solution using $O(\frac{n^2}{w \log s})$ queries [ADKF70, Wil07]. Such a large word size naturally occurs in settings such as external memory models.

Intuitively, one would guess that the true complexity of the VMV problem is actually $\frac{n^2}{w \log s}$. However, it is a major open problem in the field of data structures to prove a lower-bound for *any* static data structure problem where the number of queries is shown to be $\omega(\log |\mathcal{Q}|)$. We *do not* solve that open problem in this paper. Indeed, it is well known that any purely communication complexity based approach, such as ours and most past techniques, is doomed to give bounds at best $\Theta(\log |\mathcal{Q}|)$. What we do develop is a novel general technique for establishing strong lower bounds, that are also the best possible using communication complexity method alone, for natural 1-bit output problems based on matrix-vector multiplication. Previous techniques do not seem to yield such bounds for this important class of problems.

In their seminal paper, Miltersen et al. [MNSW98] study the span problem, where we need to store a vector space V and decide, given a query y, whether $y \in V$. This is equivalent to determining whether $x \cdot y = 0$ if the matrix x is chosen so that $V = \ker x$; i.e. we want to know if the number

[2] Note that the success probability of $\frac{2}{3}$ is achievable by random guessing.

of rows of x orthogonal to y is n, or not. For this problem, [MNSW98] show lower bound on the number of queries, similar to our own, but in the randomized setting with just one-sided constant error — the data-structure scheme is allowed to err only when $x \cdot y = 0$.

When first thinking about the VMV problem, one soon realizes that there is a one-sided error randomized reduction from the span problem to the VMV problem, and that this might be enough to give us a one-sided error lower-bound to the VMV problem.[3] But, it turns out, the error of the reduction is on the wrong side, and this does not allow us to derive any lower-bound for VMV from the [MNSW98] lower-bound for the span problem. To our knowledge, our randomized lower-bound for VMV is also the first *deterministic* lower-bound for VMV.

Note that this one-sided error lower-bound of Miltersen et al. for the span problem immediately implies the same lower-bound for the OVC problem (although not for the OGMaj problem); however, it can be shown that there is a *two-sided error* randomized data-structure scheme for the span problem where the number of queries is $O(\frac{n}{w})$, and this implies that our randomized lower-bounds for OVC cannot possibly work for the span problem.

☞ Rather remarkably, this difference between the span problem and counting orthogonal vectors may be explained by the fact that the randomized parity decision-tree complexity of the (negated) Boolean OR function is $O(1)$, but is $\Omega(n)$ for the mod-3 function. To understand why this is relevant, we need to make a detour into asymmetric communication complexity, and explain how Theorem I is proven.

1.2 Our tool: an asymmetric simulation theorem

To prove our data-structure lower bounds of Theorem I, we develop a technique of independent interest for proving lower bounds on asymmetric communication complexity. The *asymmetric* setting is distinguished from the *usual* setting of two-party communication complexity by the following:

- One player's input is much larger than the other player's.
- The two players have different communication budgets, so we may talk about $[a, b]$-protocols where Alice communicates $\leq a$ bits and Bob communicates $\leq b$ bits. Typically the player with the large input has a higher budget.
- Only one of the players needs to learn the output, typically the player with the smaller input. This makes a difference, for example, when the task is to compute a function with an output which is larger than the communication budget.

Asymmetric communication complexity was introduced explicitly by Miltersen [Mil94],[4] and later studied more systematically in the work of Miltersen et al. [MNSW98]. In both these works, it was also shown that a lower-bound for a communication problem in this setting implies a similar lower-bound for the corresponding data-structure problem. All our lower-bounds are based on this relationship. While asymmetric communication complexity was primarily motivated by its application to proving lower bounds for data-structures [PT06, PT09, JKKR04, Pat11] and streaming algorithms [BIPW10, Woo14], it is indeed a communication model of independent interest (see for example [RR15]). Despite the significant interest, there were very few general techniques developed for proving lower bounds in this model. Two such techniques appeared in the original work of

[3] The reduction is simple and works in the communication setting: in order to know if $x \cdot y = 0$, Alice and Bob use a protocol for VMV to compute $q \cdot x \cdot y$ for a shared random vector q; if $x \cdot y = 0$, then also $q \cdot x \cdot y = 0$, but if $x \cdot y \neq 0$, then $q \cdot x \cdot y = 1$ with probability exactly $\frac{1}{2}$ over the choice of q. Alas, the reduction may err precisely in the case when $x \cdot y = 1$, so the lower-bound of [MNSW98] does not apply.

[4] However the notion appears implicitly in earlier work [Ajt88, Xia92].

Miltersen et al. The first is the Richness Method for primarily proving deterministic and randomized, one-sided-error lower-bounds. The second is the round-elimination technique for two-sided error protocols, that gives strong bounds only when the number of rounds involved is quite limited. Other techniques developed are more ingenious and problem specific, like the tour de force of Patrascu [Pat11] for proving strong bounds on lopsided Disjointness. In this work, we develop a novel and reasonably widely applicable technique that yields strong lower bounds for randomized complexity even with unrestricted number of rounds of communication. Moreover, we exhibit a function for which our technique provides strong deterministic lower bounds that the Richness Method provably cannot yield.

Our technique is based on a recent trend seen in *symmetric* communication complexity, of proving *lifting theorems*, sometimes known as *simulation theorems*. Such theorems show, for some carefully chosen two-player function $g(x; y)$, called the *gadget*, that the communication complexity of a composed function $f \circ g = f(g(x_1; y_1), \ldots, g(x_p; y_p))$, under some setting, is proportional to a corresponding measure of complexity on f multiplied by the communication complexity of g.

For example, in the paper [GPW15], building on the work of [RM99], the authors have shown that — taking the gadget g to be the indexing function — the deterministic communication-complexity of $f \circ g$ equals, up to constant factors, to the deterministic query-complexity of f times $\log n$, and used this to show a separation between the deterministic communication complexity and the partition number, which was a longstanding open problem at the time. This result was improved in a recent work of the authors [CKLM17], and independently by [WYY17]. Lifting theorems, by now, have numerous other applications, such as monotone-circuit lower-bounds [KW90, RM99, Joh01, GP14, RPRC16, Sok17], small-depth circuit lower-bounds [She09, Cha07], proof-complexity lower-bounds [BHP10, HN12], and separations of complexity classes in communication complexity [DPV09, GPW15, GLM+15, GPW17]. Many of these developments have happened recently and indeed, in FOCS 2017, a workshop [MP17] was devoted entirely to such results and their applications.

In this work, we prove two simulation theorems — a deterministic simulation theorem and a randomized simulation theorem. Our gadget is the matrix-vector product ($\mathsf{MVP}_{p\times n}$), so Alice gets a $p \times n$ matrix x, and Bob gets a single n-bit vector y, and we ask them to compute $F(x; y) = f \circ \mathsf{MVP}_{p\times n}(x, y) = f(x \cdot y)$, where f is a function of p bits.[5]

It is easy to see that this can be done with $O(d \cdot n)$ bits of communication from Alice, and $O(d)$ bits from Bob, where d is the smallest depth of a parity decision-tree (PDT) for f. If the PDT is randomized, we get a randomized protocol, if the PDT is deterministic, we get a deterministic protocol. To simulate a parity query $q \cdot (x \cdot y)$, Alice sends $q \cdot x \in \{0, 1\}^n$ to Bob, and Bob then replies with $(q \cdot x) \cdot y \in \{0, 1\}$.

Our simulation theorems show that this relatively naive protocol is, indeed, optimal up to constant factors.

Theorem II (Main Tool)**.** *Let n, $p \leq m = \frac{n}{1000}$ and $C < \frac{m}{100}$ be natural numbers and let $f : \{0, 1\}^p \to \mathcal{Z}$ be an arbitrary (possibly partial) function. Consider communication protocols where Alice gets an input $x \in \{0, 1\}^{p\times n}$ and Bob gets an input $y \in \{0, 1\}^n$.*

(a) If there exists a deterministic two-player $[C \cdot n, C]$-protocol for computing $f \circ \mathsf{MVP}_{p\times n}(x, y)$, then there exists a deterministic parity decision-tree which on input z outputs $f(z)$, and makes $\leq 40 \cdot C$ parity queries to z.

[5] Lifting theorems are generally proven for a symmetrically composed function $f \circ g^p$ which is defined as $f \circ g^p(x_1, \ldots, x_p; y_1, \ldots, y_p) = f(g(x_1, y_1), \ldots, g(x_p, y_p))$. The matrix-vector product can bee seen as an asymmetric composition, i.e. $f \circ g^{p\times 1}$, defined as $f \circ g^{p\times 1}(x_1, \ldots, x_p; y) = f(g(x_1, y), \ldots, g(x_p, y))$, where g is the inner-product function. This is more subtle because in the asymmetric composition case, all the x's participate with the same y. Although previous lifting theorems have been proven with asymmetric budgets [e.g. Joh01], ours is the first lifting theorem to work with an asymmetric composition.

(b) If there exists a randomized two-player $[C \cdot n, C]$-protocol for computing $f \circ \mathsf{MVP}_{p \times n}(x, y)$ with success probability ρ, then there exists a randomized parity decision-tree which on input z outputs $f(z)$ with success probability $\geq \rho - 2^{-m}$, and makes $\leq 200 \cdot C$ parity queries to z.

A few remarks are in order. First, Theorem II is also the first instance of *any* simulation theorem extracting a randomized PDT from a randomized communication protocol. Second, for deterministic protocols in the symmetric two party and multiparty settings for XOR functions, Hatami et al. [HHL16] and Yao [Yao15] do prove theorems lifting parity decision-tree complexity. But both results incur polynomial loss in the process of lifting. To the best of our knowledge, Theorem II *is* the first lifting theorem that characterizes parity decision-tree complexity so tightly — up to constant factors. On the other hand, the gadget size in [HHL16, Yao15] are constant whereas our gadgets are polynomially large w.r.t to the arity of the outer function f. Obtaining such tight simulation theorems, w.r.t. decision tree complexity measures in general as in Theorem II, with constant gadget size is a fundamental open problem in communication complexity.

☞ We will then prove the data-structure lower-bounds (b) and (c) of Theorem I by showing lower-bounds against randomized parity decision-trees. We will show that the randomized parity decision-tree complexity of the mod-3 function is high, and it easily follows from the work of [CR12, Vid12, She12] that the randomized parity decision-tree complexity of gap-majority is high as well. However, the randomized PDT complexity of (negated) OR is $O(1)$, which is what prevents our lower-bound from applying to the span problem mentioned above.

The lower-bound for the VMV problem — Theorem I (a) — does not directly follow from the above simulation theorems. Instead, it is proven by a *simulation-type* argument: one shows that a short protocol for the VMV problem would give us a parity decision-tree for solving a certain task, and then show that this task cannot be solved efficiently.

The proof of our simulation theorems is inspired by several previous works, most notably the recent work of Göös, Pitasi and Watson [GPW17]. However the peculiarities of the asymmetric setting call for substantial development of more ideas. In particular, we make use of a novel notion, which we call *linear* min-entropy, and of a variant thereof, which we call *smooth* linear min-entropy. We believe these two notions are interesting in their own right, and should find other uses. Implementing the simulation theorems using these notions requires delicate technical work. These are the main technical contributions of this submission.

1.3 Beating the Richness Method

The Miltersen et al. [MNSW98] paper presented two techniques for proving lower-bounds in the asymmetric settings — the richness technique [see also PT09], and the round-elimination technique [later improved by SV08].

The round-elimination technique method only works in situations where the number of rounds is small — typically sub-logarithmic. To the authors' knowledge, the Richness technique is essentially the only general method known for proving deterministic unbounded-round lower-bounds in the asymmetric setting. Even those lower-bounds which are proven in the two-sided error randomized asymmetric setting — lower-bounds such as [Pat11], which cannot be shown by the richness technique because it is limited to proving one-sided error lower-bounds — the same lower-bound (up to constant factors) can be shown in the deterministic setting using the richness technique.

Given this state of affairs, it would be tempting to think, for example, that a deterministic (or one-sided error) lower-bound for the VMV problem might exist which completely circumvents our approach based on simulation theorems. However, this might actually not be the case: we show that, at least in some situations, our simulation theorem proves a deterministic lower-bound which cannot be proven by the richness technique of Bro Miltersen et al.:

Theorem III. *There exists a promise problem $F : \{0,1\}^{p\times n} \times \{0,1\}^n \to \{0,1\}$ such that:*

- *Theorem II (a) implies that any deterministic $[a,b]$-protocol for F has $a = \Omega(n^2)$ or $b = \Omega(n)$;*
- *However, F has a randomized zero-error $[O(n), O(1)]$-protocol.*

Since any lower-bound proven by the richness technique also gives a lower-bound against randomized protocols with one-sided error (and thus zero-error), it follows that the above lower-bound cannot be proven via the richness method — it is the first known lower-bound in deterministic (unbounded-round) asymmetric communication complexity for which this is the case.

2 Overview of our techniques

All our lower-bounds follow from the well-known connection between data structures and communication complexity, which first explicitly appeared in [Mil94]: if we have a data-structure scheme for $f(x;y)$, then we obtain a protocol for the communication problem where Alice gets the data x, Bob gets the query y, and they must communicate to compute $f(x;y)$. Hence we will prove the lower-bounds for data structures of Theorem I, by proving lower-bounds for asymmetric communication problems.

In turn, our communication complexity lower-bounds are all shown by first proving a lower-bound against parity decision-trees, and then *lifting* these lower-bounds to communication complexity, by use of Theorem II, which is the main technical contribution of this paper. We will thus begin by sketching the proof of Theorem II in Section 2.1; we then sketch the proofs of the data-structure lower-bounds in Section 2.2. We made an effort to include the full proof of at least one theorem within the 10-page limit. We opted for Theorem III, whose full proof appears in Section 5.2.

2.1 Proving Theorem II

To explain how we prove our simulation theorems, it is worthwhile to give a general overview of how previous simulation theorems have been proven — the discussion broadly applies to all of [RM99, GPW15, GLM+15, CKLM17, WYY17, Wat17, GKPW17, AGJ+17] and [GPW17].

We are given a protocol for a composed function $f \circ g$ — g takes a pair (x,y) of inputs and produces a p-bit string, which is then fed to f. We wish to construct a decision-tree for computing $f(z)$ when given query access to z. The general strategy is to find a leaf in the protocol tree where z is represented, meaning that the rectangle $A \times B$ associated with said leaf is such that $z \in g(A \times B)$; this way, we may output the label which the protocol assigns to that rectangle, and it should equal $f(z)$. In the randomized case we will actually want a specific distribution on such rectangles, but let's set that aside for now.

So we go down the protocol tree, keeping in mind a rectangle $A \times B$. As long as we haven't queried z, we need to make sure that *every* z is represented in $g(A \times B)$; once we have made *some* queries to z, then every z' which is consistent with those queries must be represented in $g(A \times B)$.

If the gadget g is well-chosen, it becomes feasible to enforce this invariant. For example, if $g = (\mathsf{IP}_n)^p$ is the p-fold inner-product of n-bit strings,[6] there are two known properties which, if true of A and B both, ensure that every z is represented in $g(A \times B)$ — one such property is called *thickness* and is used in [RM99, GPW15, CKLM17, AGJ+17], and another is called *density*, and is used in [GLM+15, GPW17, Wat17, GKPW17]. It is worthwhile to briefly review these notions.

For $\delta \in [0,1]$, a set $A \subseteq \{0,1\}^{p\times n}$ is called δ-*thick*, if for every $a = (a_1, \ldots, a_p) \in A$ and every $i \in [p]$, there exist $\geq 2^{\delta n}$-many different a'_i such that $(a_1, \ldots, a_{i-1}, a'_i, a_{i+1}, \ldots, a_p) \in A$; A is called δ-*dense*,

[6] Note that, g here is a function which outputs a p-bit string. We maintain this convention through out the paper.

if for every $I \subseteq [p]$ of size $|I| = k \geq 1$, the distribution $(\mathbf{x})_I$, obtained by picking a uniformly-random $\mathbf{x} \in A$ and projecting onto the coordinates in I, has min-entropy $\geq \delta\ kn$.

The *thickness* of A is then the largest δ for which it is δ-thick, and the *density* of A is the largest δ for which it is δ-dense. We may also say that A is δ-thick or δ-dense with respect to a set $S \subseteq [p]$ of coordinates, if we replace $[p]$ with S in the above definitions.

In order to find the desired leaf in the protocol tree, and thus prove the simulation theorem, the decision tree goes down the protocol tree while being careful to preserve one such property (density or thickness) as an invariant. As the rectangle becomes smaller, and we are at risk of loosing our invariant, we must have a means of restoring it by querying some coordinates of z. We then focus only on those inputs (x, y) such that $g(x, y)$ is consistent with the outcome of these queries, and it is important that our property (e.g. thickness or density) still holds with respect to those coordinates which we did not query yet.

All of the simulation theorems just mentioned follow this general pattern, and so do the simulation theorems proven in this paper. But, even after having a good understanding of this general framework, it is not *apriori* clear how to proceed when the inner gadget is the matrix-vector product, nor how to connect such results to the vector-matrix-vector problem which is not itself a composed function.

Let g be the matrix-vector product over $\mathbb{F}_2$, so that $g(x, y) = x \cdot y$ where x is a $p \times n$ matrix and y is an n-bit vector. The first thing to observe, when using g as a gadget, is that if Alice has a matrix $x \in \{0,1\}^{p \times n}$ and Bob a vector $y \in \{0,1\}^n$, then they are able to make a "parity query" to $g(x, y) = x \cdot y \in \{0,1\}^p$ by having Alice send only n bits and Bob send only 1 bit: to compute $q \cdot x \cdot y$, Alice sends over $q \cdot x \in \{0,1\}^n$, and then Bob computes and returns $(q \cdot x) \cdot y \in \{0,1\}$. So it follows that the communication complexity of $f(x \cdot y)$ is upper-bounded by the randomized parity decision-tree complexity of f.

This seems to make the properties of density and thickness unsuitable for carrying out the above strategy. Indeed, it is easy to construct, for example, a dense set A such that $g(A \times \{0,1\}^n)$ is missing some vectors — indeed, if A is the set of matrices such that the bitwise XOR of all the rows is the zero vector, then every $g(A \times \{0,1\}^n)$ is missing all vectors with odd Hamming weight. On the other hand, thickness is a property which is difficult to preserve, and it would seem that if we were able to preserve this property as an invariant in our construction, we would obtain a simulation theorem for normal decision trees, not parity decision trees. However, we cannot obtain such a simulation from the above protocol for making a "parity query" to $x \cdot y$. So we had to devise a different property for our invariant. We call it *linear min-entropy*.

Definition. The *linear min-entropy* of a set A of $p \times n$ matrices is the maximum $\eta \in [0, 1]$ such that, for every $k' \times p$ matrix Q', the distribution $Q' \cdot \mathbf{x}$ — obtained by picking a uniformly random $\mathbf{x}$ in A, and then outputting the product $Q' \cdot \mathbf{x} \in \{0,1\}^{k' \times n}$ — has min-entropy $\geq \eta\ k'n$.

So, in some sense, we require a certain min-entropy from the linear combinations of the rows of a random matrix from A. We will also need to look at a variant of this notion, called *smooth linear min-entropy*, which is the maximum linear min-entropy among all subsets $A' \subseteq A$ which preserve all but an exponentially-small fraction of A.

As one may see, it is a property stronger than density, as one demands a lower-bound on the min-entropy of *any* linear combination of coordinates, and not just of the coordinates themselves.

It will then happen that if A has linear min-entropy at least $\frac{4}{5}$, say, and $|B| \geq 2^{\frac{9}{10}n}$, then every $z \in \{0,1\}^p$ is represented in $g(A \times B)$. We will show something even stronger, a result which we call *pruning lemma*: for any such A and B, we may remove an exponentially-small fraction of A and B, to obtain a subrectangle $A' \times B' \subseteq A \times B$, such that every z appears in every row and column of the $g(A' \times B')$ communication matrix[7] in roughly equal proportion. Meaning every row and every

[7] I.e., the matrix with rows indexed by A', columns indexed by B', and with the (x, y) entry equal to $x \cdot y$.

column of the $g(A' \times B')$ communication matrix will be (roughly) equally split among the different $z \in \{0,1\}^p$.

The pruning lemma is then used to show a result called *entropy-restoring partition*. It can be considered the heart of the proof of the simulation theorems in this paper. This result shows how one may take a set $A \subseteq \{0,1\}^{p \times n}$, such that the smooth linear min-entropy of A is not too high ($\leq \frac{9}{10}$), but where the linear min-entropy of A is still somewhat high ($\geq \frac{4}{5}$), and partition A into subsets $A^\dagger, A_1, A_2, \ldots$, with $A^\dagger$ very small, such that in each A_i we have fixed some linear combination of rows (of the matrices in A_i), and where each A_i has large linear min-entropy ($\geq \frac{9}{10}$) on the remaining (linearly-independent) linear combinations of the rows. Furthermore, if we have a large set $B \subseteq \{0,1\}^n$ of vectors ($|B| \geq 2^{\frac{9}{10}n}$), we may do this in a way that for each $x \in A_i$, the values of $x \cdot y$, for $y \in B$ are equidistributed among the various possible $z \in \{0,1\}^p$ — this means that when the linear decision-tree queries the k coordinates of z corresponding to the rows which were fixed, B will be cutoff by no more than 2^{-k}. The full statement appears in Section 4.4. This is the main technical device which allows us to maintain a rectangle $A \times B$ where A has large linear min-entropy, and B is large, as we go down the protocol tree in our simulation theorem. On its own, the entropy-restoring partition suffices for proving our deterministic simulation theorem — Theorem II (a); in fact, the existence of a single part A_1 of the entropy-restoring partition is enough for the deterministic simulation theorem, whereas the randomized simulation theorem needs the full entropy-restoring partition.

To prove the randomized simulation theorem, Theorem II (b), we will use a crucial insight from [GPW17]. Suppose $\bar{\pi}$ is a randomized protocol for $f \circ g$ which is the convex combination of several deterministic protocols π. Then a good approach to proving a randomized simulation theorem is the following: in order to obtain a decision-tree for f, it suffices to be able to approximate, for each deterministic protocol π, the distribution $\pi^{-1}(z)$ obtained by running π on a random input (x,y) such that $g(x,y) = z$. We want to do this by making few queries to z, and for this purpose [GPW17] proves a result called the *inverse-marginals lemma*. Our version of this lemma states that if A has large linear min-entropy and B is large, then for any $z \in \{0,1\}^p$, if we choose a uniformly random $(x,y) \in A \times B$ among those such that $x \cdot y = z$, then the x-marginal will be close to a uniform distribution on A and the y-marginal will be close to a uniform distribution on B.

To illustrate how this is used, suppose that we are simulating π on a rectangle $A \times B$, and it was Alice's turn to communicate, and she would send bit b when $x \in A_b$ — for the partition $A = A_0 \cup A_1$; then if one were to pick a uniformly-random input in $g^{-1}(z) \cap A \times B$, then Alice would send $b = 0$ with probability roughly $\frac{|A_0|}{|A|}$ and send $b = 1$ with probability roughly $\frac{|A_1|}{|A|}$. This heuristic allows us to construct a randomized parity decision-tree which will produce, on input $z \in \{0,1\}^p$, a transcript of the protocol which is exponentially close, in statistical distance, to the transcript which we would obtain if we had run the protocol on a uniformly-random input from $g^{-1}(z)$ — which is enough to prove Theorem II (b).

2.2 The data-structure lower-bounds

The data-structure lower-bounds (b) and (c) of Theorem I follow from lower-bounds against randomized parity decision-trees, by using Theorem II (b) and the connection between data structures and asymmetric communication complexity.

It is intuitive that counting mod-3 should be hard for parity decision-trees. This is shown in Lemma 5.17 of the paper, by making use of the *polynomial discrepancy lemma* of [Cha07]. The polynomial discrepancy lemma says that the Mod3 function (roughly) equally splits the zero set of any linear form over $\mathbb{F}_2$.[8] This will imply that any randomized parity decision-tree for Mod3 will

[8] Indeed, the lemma holds for any low-degree polynomial over $\mathbb{F}_2$, not just linear forms, hence the name *polynomial*

succeed with probability $\leq \frac{1}{3} + 2^{-\Omega(n)}$. If counting mod-3 is hard, then so is counting in general, which gives us the lower-bound for OVC — Theorem I (b).

By way of binary search we can use a single majority $\log n$ times to count exactly. This would easily give us I (c), but with a $\log n \cdot \log\log n$ factor loss. However it follows from [CR12, Vid12, She12] that the randomized parity decision-tree complexity of $\sqrt{n}$-gap-majority is $\Omega(n)$, and this implies Theorem I (c).

The data-structure lower-bound of Theorem I (a) does not seem to follow directly from a lower-bound on randomized PDTs for some function. The VMV problem is quite different to a composed problem — in a composed problem both players know the outer function f and the lower-bound depends on f having large PDT complexity; we may think of the VMV problem as if only Bob knew the outer function — q is a parity which is given as Bob's input. But we can still prove the lower-bound by an interesting analogy: instead of proving a lower-bound for randomized PDTs trying to compute a certain function, we instead prove (in Lemma 6.1 of the paper) a lower-bound for randomized PDTs trying to succeed at the following task:

Lemma (Impossible task)**.** *Suppose we have a randomized parity decision-tree running in time t which, on every input $z \in \{0,1\}^p$, outputs a pair $(q, b) \in \{0,1\}^p \times \{0,1\}$ such that both:*

- *q is (always) linearly-independent of the set Q of parity queries made, and*
- *with probability ρ over the choice of q, we have $q \cdot z = b$.*

Then either $t \geq p$ or $\rho = 1/2$.

Then, analogously to the simulation theorem — Theorem II (b) — we prove (in Theorem 6.2 of the paper) that any randomized communication protocol for VMV, succeeding with probability ρ, would give us a randomized parity decision-tree for the above task, succeeding with probability $\geq \rho - 2^{-\Omega(n)}$. This establishes a lower-bound on the asymmetric randomized communication complexity of VMV, which then gives us Theorem I (a).

2.3 Beating the richness method

Theorem III is also obtained by lifting a randomized parity decision-tree lower-bound to a communication lower-bound, using Theorem II (a). The problem being lifted is a canonical promise problem having small zero-error randomized query complexity, but large deterministic query complexity.

Let $Z^{-1}(0) \subseteq \{0,1\}^{2n}$ be the set of binary strings which have $z_i = 0$ whenever i is odd, and $z_i = 1$ for at least $\frac{n}{10}$-many even coordinates i. Let $Z^{-1}(1) \subseteq \{0,1\}^{2n}$ have instead $z_i = 0$ whenever i is *even*, and $z_i = 1$ for at least $\frac{n}{10}$-many *odd* coordinates i; let $Z : \{0,1\}^{2n} \to \{0,1\}$ be the corresponding promise problem.

It is very easy to see that Z has an $O(1)$-query zero-error randomized decision-tree, and that its deterministic query-complexity is $\Omega(n)$. As it turns out, it also has deterministic pdt complexity $\Omega(n)$. This is proven in Lemma 5.5 of the paper, using an argument analogous to the Hamming bound of coding theory.

Our promise problem F is then $Z \circ g$ where $g(x; y) = x \cdot y$ is the matrix-vector product gadget. On one hand, our deterministic simulation theorem shows that F has no deterministic protocol where Alice sends $o(n^2)$ bits and Bob sends $o(n)$ bits; on the other hand, F will have a randomized zero-error protocol where Alice sends $O(n)$ and Bob sends $O(1)$ bits — hence any lower-bound for F using the richness method will fail to show that Alice must send $\omega(n)$ bits.

---

discrepancy lemma.

2.4 Outline

We work on various preliminaries in Section 3, building up to the definition of smooth linear min-entropy in Section 3.4. In Section 4.3, we prove the pruning lemma, the entropy-restoring partition, and the inverse-marginals lemma. The deterministic and randomized simulation theorems are proven in Section 5. As direct application of these theorems, we show lower-bounds on data-structures for counting orthogonal vectors (Section 5.4), and we show a lower-bound beating the richness method (Section 5.2). Finally, in Section 6 we show how a variant of the technique for proving the randomized simulation theorem gives us randomized lower-bounds against data-structures for the vector-matrix-vector problem.

3 Notation and preliminaries

Notation Throughout the paper, we will usually use capital letters to denote sets, Greek letters to denote real numbersuse, and bold-face letters to denote random variables. We will use λ to denote the empty string, and $\varnothing$ to denote the empty set.

We assume the reader is comfortable with communication complexity [KN97].

3.1 Functions of interest

The inner-product function on n-bits, denoted IP_n in defined on $\{0,1\}^n \times \{0,1\}^n$ to be:

$$\mathsf{IP}_n(x,y) = \sum_{i\in[n]} x_i \cdot y_i \bmod 2.$$

The $p \times n$ matrix-vector product function $\mathsf{MVP}_{p\times n} : \{0,1\}^{p\times n} \times \{0,1\}^n \to \{0,1\}^p$ is:

$$\mathsf{MVP}_{p\times n}(x,y) = x \cdot y = \left(\sum_{j=1}^{n} x_{ij} y_j\right)_{i\in[p]}$$

The vector-matrix-vector product function $\mathsf{VMV}_{p\times n} : \{0,1\}^{p\times n} \times (\{0,1\}^p \times \{0,1\}^n) \to \{0,1\}$ is given by:

$$\mathsf{VMV}_{p\times n}(x;q,y) = q \cdot x \cdot y = \sum_{i=1}^{p}\sum_{j=1}^{n} q_i x_{ij} y_j$$

The mod-3 function $\mathsf{Mod3}_n : \{0,1\}^n \to \{0,1,2\}$ is given by

$$\mathsf{Mod3}_n(x_1,\ldots,x_n) = \sum_{i=1}^{n} x_i \mod 3$$

For each $a \in \{0,1,2\}$, define also the function $\mathsf{Mod3}_n^a : \{0,1\}^n \to \{0,1\}$:

$$\mathsf{Mod3}_n(x_1,\ldots,x_n) = [\mathsf{Mod3}(x) = a] = \begin{cases} 1 & \text{if } \sum_{i=1}^n x_i = a \mod 3 \\ 0 & \text{otherwise.} \end{cases}$$

The gap-majority partial function $\mathsf{GMaj}_n : \{0,1\}^n \to \{0,1\}$ is given by

$$\mathsf{GMaj}_n(x_1,\ldots,x_n) = \begin{cases} 1 & \text{if } \sum_{i=1}^n x_i \geq \frac{n}{2} + \sqrt{n}, \\ 0 & \text{if } \sum_{i=1}^n x_i \leq \frac{n}{2} - \sqrt{n}. \end{cases}$$

3.2 Interval algebra

We will use the following notation to denote closed intervals of the real line:

- If δ is a non-negative real, $1 \pm \delta$ denotes the interval $[1-\delta, 1+\delta]$.
- For two intervals $I = [a, b]$ and $J = [c, d]$, $IJ = \{xy \mid x \in I, y \in J\}$, $I + J = \{x + y \mid x \in I, y \in J\}$, and if $0 \notin J$, then $\frac{I}{J} = \{\frac{x}{y} \mid x \in I, y \in J\}$.
- For an interval $J = [a, b]$ and $x \in \mathbb{R}$, $xJ = \{xy \mid y \in J\}$, $x + J = \{x + y \mid y \in J\}$ and (if $0 \notin J$) $\frac{x}{J} = \{\frac{x}{y} \mid y \in J\}$.

The following is easy to verify:

Proposition 3.1. *Let $0 \leq \delta < 1/2$ and x, y be reals.*

- *(Weak inverse) $\frac{1}{1\pm\delta} \subseteq 1 \pm 2\delta$.*
- *(Weak symmetry) If $x \in (1 \pm \delta) \cdot y$ then $y \in (1 \pm 2\delta) \cdot x$.*

3.3 Affine product

By linear independence of two k-bit strings, we mean linear independence over $\mathbb{F}_2^k$:

Definition 3.2 (Sets of linearly-independent vectors)**.** Let p, k, k' be positive integers. Let $Q \in \{0,1\}^{k \times p}$ be a matrix with k rows $q_1, \ldots, q_k \in \{0,1\}^p$. The rows of Q are said to be *linearly independent* if there is no subset $S \subseteq [k]$ such that the bitwise XOR $\bigoplus_{i \in S} q_i$ equals the all-zero vector.

If $Q' \in \{0,1\}^{k' \times p}$ is a matrix with k' rows $q'_1, \ldots, q'_{k'}$, then $Q \cup Q'$ is the matrix with $k + k'$ rows formed by the rows of Q and Q'.

Then Q' is said to be *independent of* Q if the rows of $Q \cup Q'$ are linearly-independent. This implies, in particular, that the rows of Q' are linearly independent.

For the next set of definitions, we view $x \in \{0,1\}^{p \times n}$ as a $p \times n$ Boolean matrix.

Definition 3.3 (Affine product)**.** Let $Q \in \{0,1\}^{k \times p}$ be a matrix with k linearly-independent rows $q_1, \ldots, q_k \in \{0,1\}^p$, and let $u \in \{0,1\}^{k \times n}$ be a $k \times n$ matrix. For any $p \times n$ matrix $x \in \{0,1\}^{p \times n}$, let $q_i \cdot x \in \{0,1\}^n$ be the matrix multiplication of q_i and x, i.e., bitwise-XOR of the rows x_j such that $q_{ij} = 1$. Let $Q \cdot x$ denote the matrix obtained by matrix-multiplication of Q with x, i.e. the $k \times n$ matrix $\{0,1\}^{n \times k}$ has rows $(q_1 \cdot x, \ldots, q_k \cdot x)$.

Then for any $u \in \{0,1\}^{k \times n}$ and $Q \in \{0,1\}^{k \times p}$, we call *$n$-dimensional affine product* of Q and u to be the set

$$\mathsf{Affine}^n(Q, u) = \{x \in \{0,1\}^{p \times n} \mid Q \cdot x = u\}.$$

For a given Q and u as above, and for a set $A \subseteq \mathsf{Affine}^n(Q, u)$, we define the (Q, u)-*density* of A to be the real value $\frac{|A|}{|\mathsf{Affine}^n(Q,u)|} \in [0, 1]$.

Observation 3.4. Consider $Q \in \{0,1\}^{k \times p}$ and $Q'' \in \{0,1\}^{k'' \times p}$ be two sets of linearly independent vectors, with Q'' independent of Q, and let $Q' = Q \cup Q''$, $k' = k + k''$, $u \in \{0,1\}^{k \times n}$, $u'' \in \{0,1\}^{k'' \times n}$ and $u' = uu'' \in \{0,1\}^{k' \times n}$. Then

$$\frac{|\mathsf{Affine}^n(Q', u')|}{|\mathsf{Affine}^n(Q, u)|} = \frac{1}{2^{k''n}}.$$

3.4 Linear min-entropy

In this subsection we define the two notions of linear min-entropy and smooth linear min-entropy, which are the crucial new concepts underlying the proofs below.

Definition 3.5. The *min-entropy* of a distribution μ, denoted $\mathsf{H}_\infty(\mu)$, is:

$$\mathsf{H}_\infty(\mu) = \min_x \log \frac{1}{\mu(x)}.$$

The min-entropy of a random variable $\mathbf{x}$ is the min-entropy of its underlying distribution.

The notion of *min-entropy* is not very flexible. A certain distribution could have small min-entropy simply because of a few outliers. This can be solved by allowing such outliers to be thrown away, thus obtaining a so-called *smooth* version of min-entropy: the ε-smooth min-entropy of μ is the largest min-entropy of any distribution μ' which is ε-close to μ in statistical distance. Such smooth notions have seen various applications in information theory [see RW04].

For our results, we will need a smooth version of a certain kind of min-entropy, which we call *linear min-entropy*. It is a linear variant of the notion of *density*, appearing in [GLM+15, GPW17, Wat17, GKPW17]: a set $A \subseteq \{0,1\}^{p\times n}$ is δ-*dense* if by taking a uniform $x \in A$, and projecting x onto any k coordinates, the resulting distribution has $\geq \delta kn$ min-entropy. This requirement is strengthened in the definition of linear min-entropy, which allows the projection, instead of being just k coordinates, to be any choice of k linear combinations.

Definition 3.6 (Smooth linear min-entropy)**.** Let $Q \in \{0,1\}^{k\times p}$ be a matrix with k linearly-independent rows $q_1, \ldots, q_k \in \{0,1\}^p$, let $A \subseteq \{0,1\}^{p\times n}$, and let $\varepsilon \in [0,1)$ be a real number.

The *ε-smooth linear min-entropy of A with respect to Q*, denoted $\mathsf{LH}^\varepsilon_\infty(A,Q)$, is the largest value $\eta \in [0,1]$ for which there there exists a subset $A' \subseteq A$ of size $|A'| \geq (1-\varepsilon)|A|$, such that every $Q' \in \{0,1\}^{k'\times p}$ which is independent of Q has

$$\mathsf{H}_\infty(Q' \cdot \mathbf{x}') \geq \eta\, k' n,$$

where $\mathbf{x}'$ is the random variable distributed uniformly over A'. We call *linear min-entropy of A* to the case when $\varepsilon = 0$, and denote $\mathsf{LH}_\infty(A,Q) = \mathsf{LH}^0_\infty(A,Q)$.

Remark 3.7. For quick reference, we will also make a note of the following cases, which follow from Definition 3.6.

- If $\mathsf{LH}^\varepsilon_\infty(A,Q) < \eta$, then for all $A' \subset A$ containing at least a $1-\varepsilon$ fraction of A, there exists $Q' \in \{0,1\}^{k'\times n}$ linearly independent of Q for which $\mathsf{H}_\infty(Q', \mathbf{x}') < \eta\, k' n$.
- If $\mathsf{LH}^\varepsilon_\infty(A,Q) \geq \eta$, then there exists an $A' \subset A$ containing at least a $1-\varepsilon$ fraction of A, such that every $Q' \in \{0,1\}^{k'\times n}$ linearly independent of Q gives $\mathsf{H}_\infty(Q', \mathbf{x}') \geq \eta\, k' n$.

Observe that any large subset of a set with high linear min-entropy also has somewhat high linear min-entropy:

Lemma 3.8. *Let $Q \in \{0,1\}^{k\times p}$ be a matrix with k linearly-independent rows $q_1, \ldots, q_k \in \{0,1\}^p$, and let $A \subseteq \{0,1\}^{p\times n}$, such that $\mathsf{LH}_\infty(A,Q) \geq \eta$. Then for all subset $A' \subseteq A$ with $\frac{|A'|}{|A|} \geq \delta$, $\mathsf{LH}_\infty(A',Q) \geq \eta + \frac{\log \delta}{n}$.*

Proof. Consider any $Q' \in \{0,1\}^{k' \times p}$ that is independent of Q, and let $\mathbf{x}$ be uniformly distributed over A. From the premise, we have $\mathsf{H}_\infty(Q' \cdot \mathbf{x}) \geq \eta\, k' n$, i.e., for any $w \in \{0,1\}^{nk'}$,

$$\Pr_{\mathbf{x} \in A}[Q' \cdot \mathbf{x} = w] \leq 2^{-\eta\, k' n}.$$

Let $A'(w) = \{x \in A' \mid Q' \cdot x = w\}$. Then the previous equation implies that $\frac{|A'(w)|}{|A|} \leq 2^{-\eta\, k' n}$. But if $A' \subseteq A$ has $\frac{|A'|}{|A|} \geq \delta$, then

$$\frac{|A'(w)|}{|A'|} \leq \frac{|A'(w)|}{\delta |A|} \leq 2^{-\eta\, k' n - \log \delta}$$ □

We will repeatedly make use of the following notation in the statement and proofs of various lemmas throughout the paper: Suppose $B \subseteq \{0,1\}^n$, $u \in \{0,1\}^{k \times n}$ and $w \in \{0,1\}^k$. Then we use $B(u,w)$ as notation for the subset:

$$B(u,w) = \{y \in B \mid u \cdot y = w\}.$$

3.5 Asymmetric communication and data structures

We may define an asymmetric communication problem as a function $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$. Alice gets $x \in \mathcal{X}$ and Bob gets $y \in \mathcal{Y}$, and they must communicate to compute $F(x,y)$:

Definition 3.9. An $[a,b]$-protocol is any two-player protocol where Alice communicates $\leq a$ bits and Bob is communicates $\leq b$ bits. A randomized $[a,b,\rho]$-protocol for an asymmetric communication problem $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ is a randomized $[a,b]$-protocol, where Alice gets $x \in \mathcal{X}$ and Bob gets $y \in \mathcal{Y}$, and with probability $\geq \rho$ after the communication Bob has succeeded in learning $F(x,y)$.

We may also define a static data-structure problem as a function $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, where $\mathcal{X}$ is the set of *data*, $\mathcal{Y}$ is the set of *queries*, and $\mathcal{Z}$ the set of *outputs*. A scheme is then a way of encoding elements $x \in \mathcal{X}$ so that it is possible to quickly find $F(x,y)$ for any $y \in \mathcal{Y}$.

Definition 3.10. A randomized $[s,w,t,\rho]$-scheme for the static data-structure problem $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ is a pair (E,τ), where $E : \mathcal{X} \to (\{0,1\}^w)^s$ is the *encoding function*, and for each $y \in \mathcal{Y}$, τ_y is a randomized decision-tree over s-long words in the $\{0,1\}^w$ alphabet, with $\mathcal{Z}$-labeled leaves[9], such that, for every $x \in \mathcal{X}, y \in \mathcal{Y}$, $\Pr[\tau_y(E(x)) = F(x,y)] \geq \rho$.

A deterministic $[s,w,t]$-scheme is a randomized $[s,w,t,1]$-scheme.

The following well-known lemma relates the communication complexity of F to the existence of data structures for F, and its converse translates lower-bounds for the former into lower-bounds for the latter.

Lemma 3.11. *If F has an $[s,w,t,\rho]$-scheme, then F has a $[tw, t \log s, \rho]$-protocol.*

From the above lemma we may derive the following:

Corollary 3.12. *If there is no $[C \cdot n, C, \rho]$-protocol for F, then any randomized data-structure scheme for F that uses s cells, each storing $w \leq n$ bits, must either make $t \geq \Omega(\frac{C}{\log \frac{sw}{n}})$ probes, or have success probability $< \rho$.*

Proof. Any $[s,w,t,\rho]$-scheme with $w \leq n$ would also give us an $[\frac{sw}{n}, n, t, \rho]$-scheme; applying Lemma 3.11 to the latter would then result in a $[tn,]$-protocol, which we assumed not to exist. □

[9] I.e., each non-leaf node in τ_y is labeled by an index $i \in [s]$ — meaning "query position i when arriving at this node" — and has 2^w-many children — one for each possible value in $\{0,1\}^w$; each leaf node is labeled by a possible output $z \in \mathcal{Z}$ — meaning "output z when reaching this node".

3.6 Tossing biased coins

Suppose we have a collection of (possibly biased) coins, and we carry out a random process where we repeatedly choose a coin from our collection and then toss it; then we choose a second coin and toss that one, and so on, in total tossing C-many coins. Each toss of the coins is an independent event, but the choice of the coins themselves isn't necessarily so, e.g. we might start by choosing a uniformly random coin from our collection, but then for our second coin we will only choose Coin X if we started by choosing coin Y and it turned up heads, and if either of these things failed to happen, we choose Coin Z instead.

Then we may model the above random process by a sequence of random variables $\boldsymbol{\beta}_1, \boldsymbol{\rho}_1, \mathbf{k}_1, \ldots, \boldsymbol{\beta}_C, \boldsymbol{\rho}_C, \mathbf{k}_C$, such that $\boldsymbol{\beta}_i$ is the probability of the i-th chosen coin turning up 0, $\boldsymbol{\rho}_i$ is a uniform real number from the $[0,1]$ interval, and $\mathbf{k}_i \in \{0,1\}$ is the outcome of the i-th coin toss, which is 0 iff $\boldsymbol{\rho}_i \leq \boldsymbol{\beta}_i$.

If $\bar{\beta} = (\beta_1, \ldots, \beta_C)$ is a sequence of coins which we may possibly obtain by our process, and $\bar{k} = (k_1, \ldots, k_c)$ is an outcome (possible or otherwise), let us denote by $P(\bar{\beta}, \bar{k})$ the probability $\Pr[\bar{\mathbf{k}} = \bar{k} \mid \bar{\boldsymbol{\beta}} = \bar{\beta}]$ that we obtain outcome $\bar{k}$ conditioned on having chosen coins $\bar{\beta}$. Then if we let $\alpha_i = \beta_i$ if $k_i = 0$ and $\alpha_i = 1 - \beta_i$ if $k_i = 1$, it is clear that

$$P(\bar{\beta}, \bar{k}) = \Pr[\bar{\mathbf{k}} = \bar{k} \mid \bar{\boldsymbol{\beta}} = \bar{\beta}] = \prod_{i=1}^{C} \alpha_i.$$

Then let us say that $\bar{k}$ is γ-unlikely with respect to $\bar{\beta}$ if $P(\bar{\beta}, \bar{k}) \leq \gamma$. Let us say that $\bar{\mathbf{k}}$ is γ-unlikely if $\bar{\mathbf{k}}$ is γ-unlikely with respect to the chosen coins $\bar{\boldsymbol{\beta}}$.

What is the probability we get a γ-unlikely output? It can certainly be greater than γ, since there might be, e.g., two different outputs which are γ-unlikely for every possible choice of coins. We may show the following general upper bound:

Lemma 3.13. *In any coin-tossing process like the one above,* $\Pr[\bar{\mathbf{k}}$ *is* γ*-unlikely*$] \leq \gamma \cdot 2^{6C}$.

The proof of this lemma uses the same trick as in Lemma 7 of [GPW17].

Proof. Define the random variable $\boldsymbol{\alpha}_i$ to equal $\boldsymbol{\beta}_i$ if $\mathbf{k}_i = 0$ and to equal $1 - \boldsymbol{\beta}_i$ if $\mathbf{k}_i = 1$. We wish to upper-bound $\Pr[\prod_{i=1}^{C} \boldsymbol{\alpha}_i \leq \gamma]$.

Let us define $\boldsymbol{\delta}_i = \boldsymbol{\rho}_i(1 - \boldsymbol{\rho}_i)$, then we always have $\boldsymbol{\delta}_i \leq \boldsymbol{\alpha}_i$: if $\boldsymbol{\rho}_i \leq \boldsymbol{\beta}_i$ then $\boldsymbol{\alpha}_i = \boldsymbol{\beta}_i \geq \boldsymbol{\rho}_i \geq \boldsymbol{\delta}_i$, and if $\boldsymbol{\rho}_i > \boldsymbol{\beta}_i$ then $(1 - \boldsymbol{\rho}_i) < (1 - \boldsymbol{\beta}_i)$ and again $\alpha_i = 1 - \boldsymbol{\beta}_i > 1 - \boldsymbol{\rho}_i \geq \boldsymbol{\delta}_i$.

It then follows that $\Pr[\prod_{i=1}^{C} \boldsymbol{\alpha}_i \leq \gamma] \leq \Pr[\prod_{i=1}^{C} \boldsymbol{\delta}_i \leq \gamma]$, and so we will upper-bound the latter, instead. This is a much easier task: crucially and unlike $\boldsymbol{\alpha}_i$, each $\boldsymbol{\delta}_i$ is independent and identically distributed. If we let $\mathbf{x}_i = -\log \boldsymbol{\delta}_i$, then the mean is

$$\mathbf{E}[\mathbf{x}_i] = \int_0^1 \left(\log \frac{1}{\rho} + \log \frac{1}{1-\rho} \right) \, \mathrm{d}\rho = \frac{2}{\ln 2}.$$

Since the $\mathbf{x}_i$ are i.i.d., we may apply the following form of Hoefding's concentration bound:

$$\Pr\left[\sum_{i=1}^{C} \mathbf{x}_i \geq \frac{2C}{\ln 2} + C \cdot t \right] \leq e^{-2Ct^2}$$

Assume that $\gamma \le 2^{-6C}$, otherwise the lemma is trivial; applying the above:

$$\begin{aligned}
\Pr\left[\prod_{i=1}^{C} \boldsymbol{\delta}_i \le \gamma\right] &= \Pr\left[\sum_{i=1}^{C} \mathbf{x}_i \ge \log\frac{1}{\gamma}\right] \\
&= \Pr\left[\sum_{i=1}^{C} \mathbf{x}_i \ge \frac{2C}{\ln 2} + C\left(\frac{1}{C}\log\frac{1}{\gamma} - \frac{2}{\ln 2}\right)\right] \\
&\le \exp\left(-2C\left(\frac{1}{C}\log\frac{1}{\gamma} - \frac{2}{\ln 2}\right)^2\right) \\
&= \exp\left(-2C\left(\frac{(\log\frac{1}{\gamma})^2}{C^2} - \frac{4}{C\ln 2}\log\frac{1}{\gamma} + \frac{4}{(\ln 2)^2}\right)\right) \\
&= \exp\left(-\frac{2(\log\frac{1}{\gamma})^2}{C} + \frac{8}{\ln 2}\log\frac{1}{\gamma} - \frac{8C}{(\ln 2)^2}\right) \\
&= \gamma^{\frac{2\log\frac{1}{\gamma}}{C} - \frac{8}{\ln 2}} \cdot e^{-\frac{8C}{(\ln 2)^2}} \\
&\le \gamma^{\frac{2\log\frac{1}{\gamma}}{C} - \frac{8}{\ln 2}} \le \gamma^{12 - \frac{8}{\ln 2}} \le \gamma
\end{aligned}$$

□

4 How to prune an imbalanced rectangle

Suppose we have a rectangle $A \times B$ associated with a node in the communication protocol of $f \circ \mathsf{MVP}_{p\times n}$. Let's assume that A is such that $A \subseteq \mathsf{Affine}^n(Q, u)$ and $\mathsf{LH}_\infty(A, Q) \ge \frac{4}{5}$, that $u \cdot y = w$ for all $y \in B$ and that $|B| \ge 2^{\frac{9}{10}n}$. We argue that we can prune $A \times B$ to obtain a very large sub-rectangle $A' \times B'$ such that the $\mathsf{MVP}_{p\times n}$-communication matrix of $A \times B$ has the following equi-distribution properties:

(a) **Column equidistribution:** Let $Q' \in \{0,1\}^{k'\times p}$ be linearly independent of Q. For any $y \in B'$ and any w', the number of $x \in A'$ such that $Q' \cdot x \cdot y = w'$ is approximately $\frac{|A'|}{2^{k'}}$.

(b) **Row equidistribution:** Let $Q' \in \{0,1\}^{k'\times p}$ be linearly independent of Q and consider any $u' \in \{0,1\}^{k'\times n}$ of the form $Q' \cdot x$ for some $x \in A'$.[10] Then for any $w'' \in \{0,1\}^{k'}$, the number of $y \in B'$ such that $u' \cdot y = w''$ is approximately $\frac{|B'|}{2^{k'}}$.

We demonstrate an explicit algorithm to do such pruning on $A \times B$, and more generally how to do so for several sets B simultaneously. In the Section 4.1 we prove a variant of Lindsey's lemma, and use it in Section 4.2 to show two equidistribution properties of $\mathsf{MVP}_{p\times n}$. We use these two properties in Section 4 to prove the correctness of our pruning algorithm. This algorithm, in turn, is used to prove the entropy-boosting partition lemma in Section 4.4, and the inverse-marginals lemma in Section 4.5.

4.1 Lindsey's Lemma

We first note a simple variant of Lindsey's well-known lemma.

Lemma 4.1 (Lindsey's Lemma)**.** *Suppose μ is a distribution over $\{0,1\}^n$ having $\mathsf{H}_\infty(\mu) \ge \frac{3}{4}n + 1$. Then, for any $w \in \{0,1\}$, there are fewer than $2^{\frac{3n}{4}}$ strings $x \in \{0,1\}^n$ with*

$$\Pr_{y\sim\mu}[\mathsf{IP}_n(x, y) = w] \notin \left(1 \pm 2^{-\frac{n}{4}}\right) \cdot \frac{1}{2}.$$

[10] Note that different x can generate the same u'.

Proof. Let X be any large set of strings, so that $|X| \geq 2^{\frac{3n}{4}-1}$. Let η denote the uniform distribution on X. Represent η and μ as vectors in $[0,1]^{2^n}$, in the usual way, and let $\mathbf{x}, \mathbf{y}$ be random variables distributed according to η and μ, respectively. Let M denote the sign matrix of IP_n, i.e., $M_{xy} = (-1)^{\mathsf{IP}_n(x,y)}$. Then:

$$|\Pr[\mathsf{IP}_n(\mathbf{x},\mathbf{y}) = 0] - \Pr[\mathsf{IP}_n(\mathbf{x},\mathbf{y}) = 1]| = |\langle \eta, M\mu \rangle| \leq \|\eta\|_2 \cdot \|M\|_2 \cdot \|\mu\|_2.$$

Where $\langle \cdot, \cdot \rangle$ is the usual inner-product (over $\mathbb{R}$), $\|\eta\|_2$ and $\|\mu\|_2$ are the L_2-norms of η and μ, and $\|M\|_2$ is M's operator norm. We now find that $\|\eta\|_2 = |X|^{-\frac{1}{2}} \leq 2^{-\frac{3n}{8}-\frac{1}{2}}$ and

$$\|\mu\|_2 = \sqrt{\sum_y \mu(y)^2} \leq \sqrt{\max_y \mu(y)} = 2^{-\frac{1}{2}\mathsf{H}_\infty(\mu)} \leq 2^{-\frac{3n}{8}+\frac{1}{2}}.$$

It is well-known and easy to show that $\|M\|_2 \leq 2^{\frac{n}{2}}$.[11] Hence $|\Pr[\mathsf{IP}_n(\mathbf{x},\mathbf{y}) = 0] - \Pr[\mathsf{IP}_n(\mathbf{x},\mathbf{y}) = 1]| \leq 2^{-\frac{n}{4}}$, and so $\Pr[\mathsf{IP}_n(\mathbf{x},\mathbf{y}) = w] \in (1 \pm 2^{-\frac{n}{4}}) \cdot \frac{1}{2}$. It then follows that the set of strings x such that $\Pr_{y\sim\mu}[\mathsf{IP}_n(x,y) = w] > \left(1 + 2^{-\frac{n}{4}}\right) \cdot \frac{1}{2}$ has size at most $2^{\frac{3n}{4}-1}$, and likewise for the set of strings x such that $\Pr_{y\sim\mu}[\mathsf{IP}_n(x,y) = w] < \left(1 - 2^{-\frac{n}{4}}\right) \cdot \frac{1}{2}$. □

4.2 The equidistribution property

The following lemma shows that a distribution on bit-strings is equidistributed if and only if every parity query on these strings is equidistributed. More precisely, it shows the non-trivial direction of this property. This property is well known and widely used [e.g. Vaz86, CG88, AGM03, Cha08, Bra11]. Below we provide a self-contained proof from first principles.

Lemma 4.2. *Suppose μ is a distribution over $\{0,1\}^m$, that $v \in \{0,1\}^{k\times m}$ is a matrix with k linearly-independent (over $\mathbb{F}_2^n$) rows $v_1, \ldots, v_k$, and that some $w \in \{0,1\}^k$ and $\varepsilon \in (0,1)$ are such that*

$$\Pr_{x\sim\mu}[v \cdot x = w] \notin (1 \pm \varepsilon) \cdot \frac{1}{2^k}.$$

Then there exists some non-empty set $S \subseteq [k]$ such that:

$$\Pr_{x\sim\mu}[v_S \cdot x = w_S] \notin \left(1 \pm \frac{\varepsilon}{2^k}\right) \cdot \frac{1}{2},$$

where $v_S = \oplus_{i\in S} v_i$ is the bitwise-XOR and $w_S = \oplus_{i\in S} w_i$ is the XOR, of the coordinates in S.

Proof. We can write the mass $\mu(\{x \mid v \cdot x = w\})$ as follows:

$$\mu(\{x \mid v \cdot x = w\}) = \sum_{x\in\{0,1\}^m} \mu(x) \cdot \prod_{i\in[k]} \frac{1 + (-1)^{v_i\cdot x + w_i}}{2} \notin (1 \pm \varepsilon) \cdot \frac{1}{2^k}$$

where the last inequality is the premise of the lemma. Expanding this, we find:

$$\sum_x \mu(x) \prod_i \frac{1 + (-1)^{v_i\cdot x + w_i}}{2} = \frac{1}{2^k} \sum_x \mu(x) \cdot (1 + \sum_{\varnothing \neq S \subseteq [k]} (-1)^{v_S \cdot x + w_S}) \notin (1 \pm \varepsilon) \cdot \frac{1}{2^k},$$

[11] Indeed,

$$(M^\perp M)_{y,y'} = \sum_x (-1)^{\mathsf{IP}_n(x,y)+\mathsf{IP}_n(x,y')} = \begin{cases} 2^{np} & \text{if } y = y' \\ 0 & \text{if } y \neq y'. \end{cases}$$

(to see the latter equality, notice that if the i-th $y_i \neq y'_i$, then by flipping x_i we can flip the summand, thus the entire sum cancels out). Then $M^\perp M = 2^n I$, and so $\|Mv\|_2 \leq \sqrt{2^n}\|v\|_2$ for any $v \in \mathbb{R}^{2^n}$.

i.e.,

$$\frac{1}{2^k} + \frac{1}{2^k}\sum_S (-1)^{w_S} \sum_x \mu(x) \cdot (-1)^{v_S \cdot x} \notin (1 \pm \varepsilon) \cdot \frac{1}{2^k},$$

Simplifying we get

$$\sum_S (-1)^{w_S} \sum_x \mu(x) \cdot (-1)^{v_S \cdot x} \notin \pm\varepsilon.$$

Since there are $2^k - 1 < 2^k$ different sets $S \neq \varnothing$, for at least one such S we must have

$$\sum_x \mu(x) \cdot (-1)^{v_S \cdot x + w_S} \notin \pm\frac{\varepsilon}{2^k},$$

which is to say

$$\mu(\{x \in A \mid v_S \cdot x = w_S\}) \notin \left(1 \pm \frac{\varepsilon}{2^k}\right) \cdot \frac{1}{2}.$$ □

Corollary 4.3. *Suppose $B \subseteq \{0,1\}^n$ and $u \subseteq \{0,1\}^{kn}$ is a matrix with k linearly-independent (over $\mathbb{F}_2^n$) rows $u_1, \ldots, u_k$, and that some $w \in \{0,1\}^k$ and $\varepsilon \in (0,1)$ are such that*

$$\Pr_{y \in B}[u \cdot y = w] \notin (1 \pm \varepsilon) \cdot 2^{-k}.$$

Then there exists some $\tilde{u} \in \mathsf{span}(u_1, \ldots, u_k), \tilde{u} \neq \bar{0}$ and a bit $b \in \{0,1\}$ such that:

$$\Pr_{y \in B}[\tilde{u} \cdot y = b] \notin \left(1 \pm \frac{\varepsilon}{2^k}\right) \cdot \frac{1}{2}.$$

Proof. Apply Lemma 4.2 with $m = n$, $v = u$ and μ uniform on B. □

Remark 4.4. Note that we cannot always hope that $u = u_i$ for some i. For example, suppose $u_i = 0^{i-1}10^{p-i}$, $w = 0^k$, and that B is the set of size 2^{n-1}:

$$B = \{y \in \{0,1\}^n \mid y_1 \oplus \ldots \oplus y_k = 1\}.$$

Then first probability (in the statement of Lemma 4.3) is 0, but the second probability is $\frac{1}{2}$ for any $u' = u_i$ (indeed for any linear combination of the u_i other than $u_1 \oplus \ldots \oplus u_k$).

Corollary 4.5. *Suppose $A \subseteq \{0,1\}^{p \times n}$ and $Q \in \{0,1\}^{k \times p}$ is a matrix with k linearly-independent rows $q_1, \ldots, q_k$, and that some $y \in \{0,1\}^n$, some $w \in \{0,1\}^k$ and some $\varepsilon \in (0,1)$ are such that*

$$\Pr_{x \sim A}\left[Q \cdot x \cdot y = w\right] \notin (1 \pm \varepsilon) \cdot \frac{1}{2^k}$$

Then there exists some $q \in \mathsf{span}(q_1, \ldots, q_k)$ and a bit $b \in \{0,1\}$ such that:

$$\Pr_{x \sim A}\left[q \cdot x \cdot y = b\right] \notin \left(1 \pm \frac{\varepsilon}{2^k}\right) \cdot \frac{1}{2}.$$

Proof. Apply Lemma 4.2 with $m = p$, $v = Q$, and μ being the distribution obtained by picking a uniformly random x from A, and outputting $x \cdot y$. □

4.3 Pruning Lemma

The following lemma is stated in the most general form which we will use. It may be simpler to read the entire lemma at first assuming $N = 1$ — this case suffices for the proof of the simulation theorems, and the case for larger N is only used in the lower-bounds for the VMV problem. Recall that the notation $B(u, w)$ has been defined in Section 3.4.

Lemma 4.6 (Pruning Lemma)**.** *Let $k < p \leq m = \frac{n}{1000}$ and $1 \leq N \leq 2^p$ be natural numbers, let $Q \in \{0,1\}^{k \times p}$ be a matrix with k linearly-independent rows $q_1, \ldots, q_k \in \{0,1\}^p$, let $w \in \{0,1\}^k$ and let $u \in \{0,1\}^{k \times n}$ be a matrix with k linearly-independent rows $u_1, \ldots, u_k \in \{0,1\}^n$.*

Let $A \subseteq \{0,1\}^{p \times n}$, and let $B_1, \ldots, B_N$ be subsets of $\{0,1\}^n$. Suppose that

(a1) $A \subseteq \mathsf{Affine}^n(Q, u)$ and $\mathsf{LH}_\infty(A, Q) \geq \frac{4}{5}$;

(a2) $\forall j \in [N]$, $B_j = B_j(u, w)$, and $|B_j| \geq 2^{\frac{9}{10}n}$.

Then there exists a subset $A' \subseteq A$, and for each $j \in [N]$ a subset $B'_j \subseteq B_j$, with the following properties being true for all $j \in [N]$:

(b1) $|A'| \geq (1 - 2^{-10m}) \cdot |A|$ and $|B'_j| \geq (1 - 2^{-10m}) \cdot |B_j|$

(b2) For all $Q'' \in \{0,1\}^{k'' \times p}$ linearly independent of Q, all $u'' \in Q'' \cdot A'$ [12], and all $w'' \in \{0,1\}^{k''}$,

$$|B_j(u'', w'')| = |B_j(uu'', ww'')| \in (1 \pm 2^{-10m}) \cdot \frac{|B_j|}{2^{k''}}.$$

(b3) For all $Q'' \in \{0,1\}^{k'' \times p}$ linearly independent of Q, all $y \in B'_j$, and all $w'' \in \{0,1\}^{k''}$,

$$|\{x \in A' \mid Q'' \cdot x \cdot y = w''\}| \in (1 \pm 2^{-10m}) \cdot \frac{|A'|}{2^{k''}}$$

It may help to think of the properties (b2) and (b3) in the following way: take any $Q'' \in \{0,1\}^{k'' \times p}$ — a set of parity queries linearly independent of Q. Now think of a matrix with rows indexed by $u \in Q'' \cdot A'$ and columns indexed by $y \in B'_j$, whose uy-entry is the product $u \cdot y \in \{0,1\}^{k''}$. Then (b2) says every row is roughly equidistributed among all possible values in $\{0,1\}^{k''}$, and (b3) says every column is also roughly equidistributed.

The following corollary is almost immediate.

Corollary 4.7. *Consider similar premise as in Lemma 4.6. Then for all $z \in \{0,1\}^p$ such that $Q \cdot z = w$, in each $A \times B_j$ there is a pair (x, y) such that $x \cdot y = z$.*

To see how this corollary follows from Lemma 4.6, let us consider any such z, and a pair (Q', w') such that $(Q \cup Q', ww')$ completely determines z — $Q' \in \{0,1\}^{(p-|Q|) \times p}$ is linearly independent of Q and $Q' \cdot z = w'$. Then, after pruning A and B_j according to Lemma 4.6, the existence of such (x, y) follows from property (b3).

[12] The set $Q'' \cdot A'$ is the set $\{Q'' \cdot x' \mid x' \in A'\}$

Proof of Lemma 4.6. A' and the various B'_j are obtained by the following pruning algorithm:

Algorithm 1 Pruning Algorithm

1: **input** A and B
2: Set $A' := A$ and $B'_j := B_j$ for all $j \in [N]$.
3: **while** there exists a vector $q \in \{0,1\}^p$ independent of Q such that $0^n \in q \cdot A'$ **do**
4: **remove** from A' every $x \in A'$ such that $q \cdot x = 0^n$.
5: **while** there exists $j \in [N]$, a vector q independent of Q, a string $\tilde{u} \in q \cdot A' \in \{0,1\}^n$, and a bit $b \in \{0,1\}$, such that $\Pr_{y \in B'_j}[\tilde{u} \cdot y = b] \notin (1 \pm 2^{-12m}) \cdot \frac{1}{2}$ **do**
6: **remove** from A' every $x \in A'$ such that $\tilde{q} \cdot x = \tilde{u}$.
7: **while** there exists a $j \in [N]$, a vector q independent of Q, a string $y \in B'_j$, and a bit $b \in \{0,1\}$, such that $\Pr_{x \in A'}[q \cdot x \cdot y = b] \notin (1 \pm 2^{-11m}) \cdot \frac{1}{2}$ **do**
8: **remove** y from B'_j.
9: **output** A' and every B'_j.

This is proven as follows:

- First notice that, at every point in the algorithm after the first while loop, every matrix $x \in A'$ has linearly-independent rows. This is because removing every q such that $0^n \in a \cdot A'$ means removing every matrix for which a linear combination of rows would result in the zero vector.

- Now let us show that (b1) holds at the end of the algorithm, i.e. that only a small fraction of the elements of A and B were pruned. Notice that as long as, say, at least $1/2$ of A and B are preserved throughout the execution of the pruning algorithm, properties (a1) and (a2) will give us that $\mathsf{LH}_\infty(A', Q) \geq \frac{3.9}{5}$ (see Lemma 3.8) and $|B'| \geq 2^{\frac{9n}{10}-1}$, i.e. A' still has large linear min-entropy and B is still large. Throughout, $\mathbf{x}'$ will denote a random element from A' and $\mathbf{y}'$ a random element from B' (note these sets change at each loop, and hence so do $\mathbf{x}'$, $\mathbf{y}'$).

  - The total number of cycles carried out in the first **while** loop above is less than 2^p; this is because the total number of q satisfying the condition is less than 2^p, and once the loop applies for a given q, this q will not appear again. For each such q, we remove a single projection from $q \cdot A'$ (i.e., all $x \in A'$ such that $q \cdot x = 0^n$, but no other); since $\mathsf{H}_\infty(q \cdot \mathbf{x}') \geq \frac{3.9}{5}n$ in each iteration, each q results in removing at most a fraction of $2^{-\frac{3.9}{5}n}$ of the elements of A', for a total $2^{-\frac{3.9}{5}n+p} \leq 2^{-\frac{77n}{100}} \ll 2^{-10m}$.
  - The total number of times we cycle the second **while** loop is less than $N \cdot 2^p \cdot 2^{\frac{3n}{4}}$; indeed, a triple $(j, \tilde{q}, \tilde{u})$ satisfying the condition (for some $\tilde{w}$) only appears once, and by Lemma 4.1 (taking μ to be uniform over B'_j) there are at most $2^{\frac{3n}{4}}$-many $\tilde{u}$ satisfying the condition. Again we conclude that the total fraction of elements of A that were removed must then be $\leq 2^{(\frac{3}{4}-\frac{3.9}{5})n+2p} = 2^{-\frac{0.6}{20}n+2p} \ll 2^{-10m}$. So, combining both the first two loops, we find that we removed (much) less than a 2^{-10m} fraction of the elements of A.
  - Here the calculation is similar. As before, the total number of cycles executed in the second **while** loop is at most $N \cdot 2^p \cdot 2^{\frac{3n}{4}}$. Indeed, for each $j \in [N]$, and each $q \in \{0,1\}^p$ independent of Q, apply Lemma 4.1 with μ being to the distribution $q \cdot \mathbf{x}'$ — this distribution has $\mathsf{H}_\infty(q \cdot \mathbf{x}') \geq \frac{4}{5}n$ since $\mathsf{LH}_\infty(A, Q) \geq \frac{4}{5}$. So let β be the total fraction of B_j that we prune; then $\beta \leq 2^{\frac{3n}{4}+p-\frac{9n}{10}} \ll 2^{-12m}$. We only needed $\beta \leq 2^{-10m}$ to satisfy property (b1) of the lemma, but we will use this tighter $\beta \ll 2^{-12m}$ bound below.

- Let us now show that property (b2), with 2^{-10m} replaced by 2^{-11m}, holds between the second and third while loops, i.e. the first time that the algorithm reaches line 7.

Suppose by contradiction that this property failed at that point. The first while loop removed all $x \in A'$ whose rows are not independent. We may then apply Lemma 4.3, with the u_i in that lemma being equal to our u_i'', and $\varepsilon = 2^{-11m}$ — the u_i are independent since the rows of $x \in A'$ are independent, and so are the rows of Q''. We can then find a non-zero $\tilde{u}$ in the span of (the rows of) u'' and a $b \in \{0,1\}$ such that (since $k'' \le p \le m$)

$$\Pr_{y \in B_j'}[\tilde{u} \cdot y = b] \notin (1 \pm 2^{-11m-k''}) \cdot \frac{1}{2} \implies \Pr_{y \in B_j'}[\tilde{u} \cdot y = b] \notin (1 \pm 2^{-12m}) \cdot \frac{1}{2}.$$

But if $\tilde{u}$ is in the span of u'', then $\tilde{u}$ is in $q \cdot A$ for some nonzero q in the span of Q'' — hence q is independent of Q. But in the second while loop of the pruning algorithm, we have eliminated all pairs of $\tilde{u}$ and q such that $\Pr_{y \in B_j}[\tilde{u} \cdot y = b] \notin (1 \pm 2^{-12m}) \cdot \frac{1}{2}$.

- Now we find that property (b2) also holds at the end of the pruning algorithm. The third while loop may remove some y's from each B_j', but we have seen that it only removes a small $\beta \ll 2^{-11m}$ fraction. So let $B^{(2)}$ be B_j' before the third while loop, and $B^{(3)}$ be B_j' after the third while loop. Then $|B^{(2)}| \in (1 \pm 2\beta)|B^{(3)}|$. But as we have just seen, for all Q'', u'' and w'', $|B^{(2)}(u'', w'')| \in (1 \pm 2^{-11m}) \cdot \frac{|B^{(2)}|}{2^{k''}}$. So we now have instead

$$\begin{aligned} |B^{(3)}(u'', w'')| &\in (1 \pm 2^{-11m}) \cdot \frac{|B^{(2)}|}{2^{k''}} \pm \beta |B^{(2)}| \\ &\subseteq (1 \pm 2^{-11m}) \cdot \frac{(1 \pm 2\beta)|B^{(3)}|}{2^{k''}} \pm 2\beta |B^{(3)}| \\ &\subseteq (1 \pm 2^{-10m}) \frac{|B^{(3)}|}{2^{k''}} \qquad \text{(since } \beta \le 2^{-12m}\text{)} \end{aligned}$$

  So we continue to maintain (b2).

- To finish, we show that (b3) holds at the end of the pruning algorithm. Suppose this was not so. Then apply Lemma 4.5, with Q in that lemma equal to our Q'' and $\varepsilon = 2^{-10m}$. We can then find a non-zero q in the span of (the rows of) Q'' and a $b \in \{0,1\}$ such that (since $k'' \le p \le m$)

$$\Pr_{x \in A'}[q \cdot x \cdot y = b] \notin (1 \pm 2^{-10m-k''}) \cdot \frac{1}{2} \implies \Pr_{x \in A'}[q \cdot x \cdot y = b] \notin (1 \pm 2^{-11m}) \cdot \frac{1}{2}.$$

  But the third while loop has eliminated all such pairs of q and b. □

4.4 Entropy-restoring partition

As we mentioned in the introduction, the entropy-restoring partition is the main technical device behind our simulation theorem. It is what allows us to maintain high linear min-entropy of Alice's set A.

Lemma 4.8 (Entropy-restoring partition)**.** *Let $k < p \le m = \frac{n}{1000}$, $1 \le N \le 2^p$ be natural numbers, and $\varepsilon = 2^{-5m}$. Let $Q \in \{0,1\}^{k \times p}$ be a matrix with k linearly-independent rows $q_1, \ldots, q_k$, let $w \in \{0,1\}^k$ and let $u \in \{0,1\}^{k \times n}$. Let $A \subseteq \{0,1\}^{p \times n}$, let $B_1, \ldots, B_N$ be subsets of $\{0,1\}^n$, and suppose that*

(a1) $A \subseteq \mathsf{Affine}^n(Q, u)$,

(a2) For all $j \in [N]$, $B_j = B_j(u, w)$ and $|B_j| \ge 2^{\frac{9}{10}n}$,

(a3) $\mathsf{LH}^{\varepsilon}_{\infty}(A,Q) < \frac{9}{10}$ *but* $\mathsf{LH}_{\infty}(A,Q) \geq \frac{4}{5}$.

Then there exist partitions $B_j = B_j^{\dagger} \cup B'_j$ *and* $A = A^{\dagger} \cup A_1 \cup A_2 \cup \ldots$, *and for every* $i \geq 1$*:*

(b1) integers $k_i = k + k'_i > k$ *and sets*

$$Q_i = Q \cup Q'_i, \qquad \text{where} \qquad Q'_i \in \{0,1\}^{k'_i \times p} \text{ is independent of } Q; \text{ and}$$

(b2) a vector $u_i = uu'_i \in \{0,1\}^{k_i n}$,

such that, for every $i \geq 1$ *and every* $j \in [N]$*, the following properties will hold:*

(c1) $|A^{\dagger}| \leq \varepsilon |A|$;

(c2) $A_i \subseteq \mathsf{Affine}^n(Q_i, u_i)$ *and*

$$\frac{|A_i|}{|\mathsf{Affine}^n(Q_i,u_i)|} \geq 2^{\frac{1}{20}k'_i n} \frac{|A|}{|\mathsf{Affine}^n(Q,u)|};$$

(c3) It holds for all $w_i = ww'_i \in \{0,1\}^{k_i}$ *that*

$$|B'_j(u_i, w_i)| \geq (1-\varepsilon)\frac{|B_j|}{2^{k'_i}};$$

(c4) $\mathsf{LH}_{\infty}(A_i, Q_i) \geq \frac{9}{10}$.

Proof. We apply the following algorithm on the sets A and B_j, to obtain the sets $A^{\dagger}, A_1, \cdots$ and B'_j. This is similar to the density-restoring partition of [GPW17][13].

Algorithm 2 Entropy-restoring Partition

input A, $B_1, \ldots, B_N$
Let A' and B'_j be obtained from A and B_j by pruning (Algorithm 1).
output B'_j and $B_j^{\dagger} = B_j \setminus B'_j$ for every $j \in [N]$.
Set $A^{\dagger} = A \setminus A'$ and $i = 1$.
while A' is such that $\mathsf{LH}_{\infty}(A', Q) < \frac{9}{10}$ and $|A'| \geq 2^{-\frac{n}{20}}|A|$ **do**
    Let $Q'_i \in \{0,1\}^{k'_i \times p}$ be a $k'_i \times p$ matrix, whose set of rows is maximal among those sets of rows which are independent of Q and such that

$$\mathsf{H}_{\infty}(Q'_i \cdot \mathbf{x}') < \frac{9}{10}$$

    (where $\mathbf{x}'$ is a random variable distributed uniformly over A').
    Let u'_i witness of this fact, so that

$$\Pr[Q'_i \cdot \mathbf{x}' = u'_i] > 2^{-\frac{9}{10}k'_i n}$$

    Output $A_i = \{x \in A' \mid Q'_i \cdot x = u'_i\}$.
    Update $A' = A' \setminus A_i$ and increment i.
if $\mathsf{LH}_{\infty}(A', Q) \geq \frac{9}{10}$ **then**
    Output $A_{i+1} = A'$.
else
    Update $A^{\dagger} = A^{\dagger} \cup A'$.
output $A^{\dagger}$

[13] If you are familiar with that paper, the following should be made clear: in [GPW17], they call "density" to a notion similar to our LH_{∞} (but without the smoothness or linearity), and the analogous concept to our $\frac{|A|}{|\mathsf{Affine}^n(Q,u)|}$ is what they call "randomness deficiency" — though again, in their paper randomness deficiency is not measured with respect to an affine subspace, but rather with respect to a product-set of binary strings.

We will have $|A^\dagger| \le 2^{-10m}|A| + 2^{-\frac{n}{20}}|A| \ll \varepsilon|A|$, giving us property (c1). By the guarantee of the Pruning Lemma 4.6, the sets B'_j satisfy property (c3) for all u_i and all w_i that we may choose. Property (c2) is given by our choice for u'_i, as follows: at line 7 of the algorithm,

$$\Pr[Q'_i \cdot \mathbf{x}' = u'_i] = \frac{|A' \cap \mathsf{Affine}(Q'_i, u'_i)|}{|A'|} > 2^{-\frac{9}{10}k'_i n};$$

since furthermore $A' \subseteq \mathsf{Affine}^n(Q,u)$, $\frac{|\mathsf{Affine}^n(Q,u)|}{|\mathsf{Affine}^n(Q_i,u_i)|} = 2^{k'_i n}$ (Observation 3.4), and $|A'| \ge 2^{-\frac{n}{20}}|A|$,

$$\frac{|A_i|}{|\mathsf{Affine}^n(Q_i,u_i)|} = \frac{|A' \cap \mathsf{Affine}^n(Q'_i,u'_i)|}{|\mathsf{Affine}^n(Q_i,u_i)|} > 2^{\frac{1}{10}k_i n} \cdot \frac{|A'|}{|\mathsf{Affine}^n(Q,u)|} \ge 2^{\frac{1}{20}k_i n} \cdot \frac{|A|}{|\mathsf{Affine}^n(Q,u)|}.$$

Property (c4) holds because of the maximality of Q'_i. Indeed, if there was Q''_i, independent of $Q \cup Q'_i$, such that $\mathsf{H}_\infty(Q''_i \cdot \mathbf{x}) < \frac{9}{10}|Q''_i|n$ where $\mathbf{x}$ is uniform over A_i, then the matrix $\tilde{Q}_i = Q'_i \cup Q''_i$ would still be independent of Q, and would have $\mathsf{H}_\infty(\tilde{Q}_i \cdot \mathbf{x}') < \frac{9}{10}(k'_i + k''_i)n$, contradicting the maximality of Q'_i among all such matrices. □

4.5 The inverse-marginals lemma

The crucial contribution of [GPW17] is the discovery and application of the so-called *inverse-marginals lemma*. This lemma says that if $A \times B$ has A "dense" and B large, then, for any $z \in \{0,1\}^p$, the uniform distribution over the set $g^{-1}(z) \cap A \times B$ has both its marginal close to uniform on A and B, where g is the p-fold indexing function. We show an analogous theorem where their notion of *density* is replaced by our notion of linear min-entropy, and the gadget g is the matrix-vector product. As in the pruning lemma above, the case $N = 1$ suffices for the simulation theorem, and the case for larger N is only needed for the VMV lower-bound.

Lemma 4.9 (Inverse-marginals lemma)**.** *Let $k < p \le m = \frac{n}{1000}$ and $1 \le N \le 2^p$ be natural numbers, let $Q \in \{0,1\}^{k\times p}$ be a matrix with k linearly-independent rows $q_1, \ldots, q_k$, let $w \in \{0,1\}^k$ and let $u \in \{0,1\}^{k\times n}$. Let $A \subseteq \{0,1\}^{p\times n}$, let $B_1, \ldots, B_N$ be subsets of $\{0,1\}^n$, and suppose that*

- *$A \subseteq \mathsf{Affine}^n(Q,u)$ and $\mathsf{LH}_\infty(A,Q) \ge \frac{4}{5}$;*
- *$\forall j \in [N]$, $B_j = B_j(u,w)$, and $|B_j| \ge 2^{\frac{9}{10}n}$.*

Suppose further that $z \in \{0,1\}^p$ is any string such that $Q \cdot z = w$, and let $(\mathbf{x},\mathbf{y})$ be random variables obtained by choosing a uniformly-random pair from the set

$$\{(x,y) \in A \times B_j \mid x \cdot y = z\}. \qquad \text{(for any fixed } j \in [N])$$

Then the marginal distribution of $\mathbf{x}$ is 2^{-8m}-close to the uniform distribution on A, and the marginal distribution of $\mathbf{y}$ is 2^{-8m}-close to uniform on B_j.

Proof. Let $A, B_1, \ldots, B_N$ be as given, and apply the Pruning Lemma (4.6) to obtain A' and B'_j (for each $j \in [N]$). Consider the following four distributions: α is uniform on A, α' is uniform on A', $\hat{\alpha}$ is the distribution of $\mathbf{x}$ above, and $\hat{\alpha}'$ is the distribution of the $\mathbf{x}'$-marginal where $(\mathbf{x}',\mathbf{y}')$ are uniformly chosen from the set $\{(x',y') \in A' \times B'_j \mid x' \cdot y' = z\}$. Then:

- $\Delta(\alpha,\alpha') \ll 2^{-9m}$ since $A' \subseteq A$ and $|A'| \ge (1 - 2^{-10m})|A|$ by property (b1) of Lemma 4.6.

- $\Delta(\alpha', \hat{\alpha}') \ll 2^{-9m}$ because $\hat{\alpha}'(x) \in (1 \pm 3 \cdot 2^{-10m})\frac{1}{|A'|}$ for every $x \in A'$. To see this, notice that (b2) of Lemma 4.6 gives us that $\hat{\alpha}'(x) \in (1 \pm 2^{-10m})\frac{1}{v}$ for the same value v; but since $\sum_{x \in A'} \hat{\alpha}'(x) = 1$, it follows that $v \in (1 \pm 2^{-10m})|A'|$, and so $\hat{\alpha}'(x) \in (1 \pm 2^{-10m})\frac{1}{(1 \pm 2^{-10m})|A'|} \subseteq (1 \pm 3 \cdot 2^{-10m}) \cdot \frac{1}{|A'|}$.

- $\Delta(\hat{\alpha}', \hat{\alpha}) \leq 2 \cdot 2^{-9m}$; this is argued as follows:
  - Let $B = B_j$, $B' = B'_j$;
  - If $x \in A$, let $B_{x,z} = \{y \in B_j \mid x \cdot y = z\}$;
  - Let $B'_{x,z} = \{y \in B'_j \mid x \cdot y = z\}$ if $x \in A'$, but set $B'_{x,z} = \varnothing$ if $x \in A \setminus A'$.
  - Let also $M_z = \sum_{x \in A} |B_{x,z}|$ and $M'_z = \sum_{x \in A} |B'_{x,z}|$.
  - We have $\hat{\alpha}(x) = \frac{|B_{x,z}|}{M_z}$ and $\hat{\alpha}(x)' = \frac{|B'_{x,z}|}{M'_z}$;
  - By property (b1), only $\frac{|B|}{2^{10m}}$ columns were removed from B; hence, for all $x \in A'$,

$$|B'_{x,z}| \leq |B_{x,z}| \leq |B'_{x,z}| + \frac{|B|}{2^{10m}}$$

    By (b2) and (b1) of Lemma 4.6, we also have

$$|B'_{x,z}| \in (1 \pm 2^{-10m})\frac{|B'|}{2^{p-k}} \subseteq (1 \pm 3 \cdot 2^{-10m})\frac{|B|}{2^{p-k}},$$

    and so (since $2^{p-k} \leq 2^m$)

$$|B_{x,z}| \in |B'_{x,z}| \pm \frac{|B|}{2^{10m}} \subseteq \left(1 \pm \frac{|B|}{|B'_{x,z}| \cdot 2^{10m}}\right) \cdot |B'_{x,z}| \subseteq (1 \pm 2 \cdot 2^{-9m})|B'_{x,z}|.$$

  - We have $|M'_{x,z}| \in (1 \pm 2^{-10m})\frac{|B'|}{2^{p-k}} \cdot |A'|$, by (b2) of Lemma 4.6; since (by b1) only $\frac{|A|}{2^{10m}}$ rows were removed from A, and only $\frac{|B|}{2^{10m}}$ columns were removed from B,

$$M'_z \leq M_z \leq (1 \pm 2^{-10m})\frac{|A'|}{2^{p-k}}|B'| + |A'|\frac{|B|}{2^{10m}} + \frac{|A|}{2^{10m}}|B|,$$

    and thus $M_z \in (1 \pm 4 \cdot 2^{-9m})\frac{|A'||B'|}{2^{p-k}}$.
  - It follows that for $x \in A'$,

$$\hat{\alpha}(x) = \frac{|B_{x,z}|}{M_z} \in (1 \pm 8 \cdot 2^{-9m})\frac{|B'_{x,z}|}{M'_z} = (1 \pm 8 \cdot 2^{-9m}) \cdot \hat{\alpha}'(x),$$

    and that

$$\sum_{x \in A \setminus A'} \hat{\alpha}(x) \leq \frac{|A \setminus A'||B|}{M_z} \leq 2^{-10m}\frac{|A||B| \cdot 2^{p-k}}{(1 \pm 2^{-10p})|B'||A'|} \leq 2 \cdot 2^{-9m}.$$

    Hence $\Delta(\hat{\alpha}, \hat{\alpha}') = O(2^{-9m})$.

It then follows by the triangle inequality that $\Delta(\alpha, \hat{\alpha}) \leq 2^{-8m}$, which shows that the $\mathbf{x}$-marginal is close to uniform on A. A similar reasoning using property (b3) instead of (b2) will show that the $\mathbf{y}$ marginal is close to uniform on B_j. □

5 Simulation theorems

The layout of this section is the following. In Section 5.1 we prove our deterministic simulation theorem, and in Section 5.3 we prove our randomized simulation theorem. In Section 5.2 we construct a certain promise problem, and use our deterministic simulation theorem to show a deterministic communication complexity lower-bound, which we also show cannot be proven via the richness method of [MNSW98].

5.1 Deterministic simulation

The following is a restatement of Theorem II (a):

Theorem 5.1 (Deterministic simulation theorem)**.** *Let n, $p \leq \frac{n}{1000}$ and $C < \frac{p}{100}$ be natural numbers. Suppose that there exists a deterministic two-player $[C \cdot n, C]$-protocol π, where Alice gets an input from $\{0,1\}^{p \times n}$ and Bob gets an input from $\{0,1\}^n$, which solves $f \circ \mathsf{MVP}_n$.*

Then there exists a deterministic parity decision-tree which on input z outputs $f(z)$; and makes $\leq 40 \cdot C$ parity queries to z.

Proof of Theorem 5.1. Abbreviate $g = \mathsf{MVP}_{p \times n}$. Suppose we are given a deterministic protocol π which solves $f \circ g$. On input $z \in \{0,1\}^p$, our decision-tree procedure τ will simulate the protocol π, in an attempt at finding a leaf of the protocol tree such that $x \cdot y = z$ for some (x, y) appearing in that leaf. If it succeeds in doing so, it then outputs the label of that leaf, which must then equal $f(z)$ by the correctness of π.

The decision-tree procedure τ appears in Algorithm 3, but before seeing it in detail let us here outline the main idea. In order to ensure that $x \cdot y = z$ for some (x, y) on the leaf that is eventually found, the decision-tree procedure of Algorithm 3 will traverse the protocol tree while keeping in mind a rectangle $A \times B$. As we are traversing node v of the protocol-tree, $A \times B$ will always be a sub-rectangle of the rectangle which π associates with v (i.e. the rectangle of inputs leading to node v), but we will have the promise that B is large, that A has high linear min-entropy with respect to the set Q of queries we already made, and that every pair $(x, y) \in A \times B$ will have $Q \cdot x \cdot y = Q \cdot z$. More precisely, the following will be enforced:

Claim 5.2. *At line 5 of Algorithm 3, the following invariants always hold:*

(i) $A \subseteq \mathsf{Affine}^n(Q, u)$, $B = B(u, w)$ *and* $Q \cdot z = w$;

(ii) $\mathsf{LH}_\infty(A, Q) \geq \frac{4}{5}$;

(iii) $|B| \geq 2^{n - C - 2|Q|}$; *and*

(iv) $|Q| \geq 40 \cdot C$

The idea of the algorithm is the following. Instead of trying to preserve large linear min-entropy of A (invariant i) — $\mathsf{LH}_\infty(A, Q) \geq \frac{4}{5}$ — we try to preserve that the *smooth* linear min-entropy of A is $\mathsf{LH}^{2/3}_\infty(A, Q) \geq \frac{9}{10}$. I.e. we give ourselves a *safety margin*. As we will see, as long as we have this safety margin, when Alice communicates there is a way of choosing this communicated bit (which will remove some elements from A) in such a way that invariant (i) will still hold. However, communicating this bit might cause us to loose our safety margin, i.e., we might get into a situation where $\mathsf{LH}^{2/3}_\infty(A, Q) < \frac{9}{10}$. In this case, we make use of the entropy-restoring partition (Lemma 4.8) to obtain a subset $A_1 \subseteq A$ and a set of queries Q'_1 such that $Q'_1 \cdot x = u_1$ for all $x \in A_1$, but where we again have our safety margin with respect to the remaining dimensions — $\mathsf{LH}^{2/3}_\infty(A_1, Q \cup Q'_1) \geq \frac{9}{10}$.

Algorithm 3 Deterministic decision-tree procedure $\tau(z)$

1: **input** $z \in \{0,1\}^p$.
2: Initialize $A = \{0,1\}^{p \times n}$, $B = \{0,1\}^n$.
3: Initialize $\varepsilon = \frac{1}{3}$, $Q = \varnothing$, $w = \varnothing$, $u = \varnothing$.
4: Let v be the root of π.
5: **while** v is not a leaf **do**

Invariants (i-iv) hold here.

6: **if** $\mathsf{LH}^{\varepsilon}_{\infty}(A, Q) \geq \frac{9}{10}$ **then**
7: Let v_0 and v_1 be the children of v.
8: **if** Alice communicates at v **then**
9: Let $A' \subseteq A$ have $\mathsf{LH}_{\infty}(A', Q) \geq \frac{9}{10}$ and $|A'| \geq (1-\varepsilon)|A|$.
10: Choose $i \in \{0,1\}$ such that $\frac{|A' \cap A_{v_i}|}{|A'|} \geq \frac{1}{2}$.
11: Update $A = A' \cap A_{v_i}$ and $v = v_i$.
12: **else if** Bob communicates at v **then**
13: Choose $i \in \{0,1\}$ such that $\frac{|B \cap B_{v_i}|}{|B|} \geq \frac{1}{2}$.
14: Update $B = B \cap B_{v_i}$ and $v = v_i$.
15: **else if** $\mathsf{LH}^{\varepsilon}_{\infty}(A, Q) < \frac{9}{10}$ **then**
16: Apply Lemma 4.8 with $N = 1$ and $B_1 = B$, to obtain

- Partitions $A = A^{\dagger} \cup A_1 \cup A_2 \cdots$ and $B = B^{\dagger} \cup B'$;
- For each $i \geq 1$, the values of k'_i, Q'_i, and u'_i.

17: **Query** z to discover $w'_1 = u'_1 \cdot z$.
18: Update $A = A_1$, $B = B'(u'_1, w'_1)$;
19: Update $Q = Q \cup Q'_1$, $u = uu'_1$ and $w = ww'_1$.
20: **Output** the label of the leaf v.

The decision tree then makes the parity queries $Q'_1 \cdot z$, and sets $B = B'(u_1, Q'_1 \cdot z)$, which will cut down Bob's set by roughly $2^{-|Q'_1|}$.

Now please inspect Algorithm 3. We will show that invariants (i-iv) are actually preserved. Since invariants (i-iv) hold when we reach a leaf, this leaf will have some (x, y) such that $x \cdot y = z$ (see Corollary 4.7), and so the output equals $f(z)$ as intended. On the other hand, exactly $|Q| \leq 20 \cdot C$ queries will be made, and so the theorem follows.

Invariant (i) is preserved by subsets in lines 11 and 14, and given by Lemma 4.6 in lines 18 and 19.

A is updated in lines 9 to 11. If A is any set such that $\mathsf{LH}^{\delta}_{\infty}(A, Q) \geq \frac{9}{10}$, then there exists a subset A' of A for which $\mathsf{LH}_{\infty}(A', Q) \geq \frac{9}{10}$ and $|A'| \geq \frac{2}{3}|A|$. The algorithm considers the bipartition of this set by Alice's communication, and goes to the *bigger* side — this side is denoted by $A' \cap A_{v_i}$. By Lemma 3.8, we are still assured that $\mathsf{LH}_{\infty}(A' \cap A_{v_i}, Q) \geq 9/10 - 1/n \geq 4/5$ — thus invariant (ii) is maintained.

A is also updated in line 18; in that line we set $A = A_1$, and in the following line we update $Q = Q \cup Q'_1$, and we are ensured by property b3 of Theorem 4.6 that $\mathsf{LH}_{\infty}(A_1, Q \cup Q'_1) \geq \frac{9}{10}$ at that point, so invariant (ii) is also preserved in this update of A.

B is updated in lines 14 and 18. If it is Bob's turn to communicate, then the algorithm again zooms into the *bigger* side induced by Bob's bipartition of B. Throughout the entire algorithm, this might result in cutting down $|B|$ by a total factor never smaller than 2^{-C}. By property b3 of

Theorem 4.6, this update of B at line 18 causes B to be cut down by a factor which is never smaller than $2^{2|Q_1'|}$, and later Q is updated to $Q \cup Q_1'$. Hence invariant (iii) is preserved.

We now show that $|Q| \leq 20 \cdot C$ throughout. For this we control the *linear density* $\frac{|A|}{|\mathsf{Affine}^n(Q,u)|}$ In the first part of the **while** loop, the density of A drops by a factor no smaller than $\frac{1}{4}$ in each iteration (line 11). There are at most C such iterations, hence this density can drop by a factor of at most 2^{-2Cn}. In the second part of the **while** loop, where parity queries are made on z and $|Q|$ increases by k' (line 16), the linear density is multiplied by $2^{\frac{1}{20}k'n}$ by property (b3) of Lemma 4.6. Since the density can be at most 1, we get $|Q| \leq 40 \cdot C$. □

5.2 A lower-bound beating the Richness Method

The *Richness-Method* of Bro Miltersen et al [MNSW98], is a method for proving lower-bounds for the communication complexity of asymmetric problems. It relies on the following definition:

Definition 5.3 (Richness)**.** A two-player problem $F : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ is said to be (u,v)-rich with respect to $z \in \{0,1\}$, if there exists $X \subseteq \mathcal{X}$ with $|X| \geq u$, such that for every $x \in X$ there exists $Y_x \subseteq \mathcal{Y}$ with $|Y| \geq v$, such that $F(x,y) = z$ for every $y \in Y_x$.

The Richness Method then consists of two steps: (a) Show that F is (u,v)-rich with respect to some $z \in \{0,1\}$. (b) Show that F does not have any z-monochromatic rectangles of size $u' \times v'$, where both $u' \geq u/2^{a+b+2}$ and $v' \geq v/2^{b+2}$. i.e., any such large rectangle must intersect $F^{-1}(1-z)$.

It will then follow that F does not have any deterministic $[a,b]$-protocols. But something stronger will then also follow: that F does not have any randomized $[a,b]$-protocols, which are allowed to err whenever $F(x,y) = z$ (for the same z for which the two properties above were shown), but not when $F(x,y) \neq z$. I.e., any lower-bound proven using the richness method will give a one-sided-error lower-bound. This follows from the cellebrated *Richness Lemma*:

Lemma 5.4 (Richness Lemma [MNSW98])**.** *Let F be a (u,v)-rich problem with respect to z. If F has a randomized one-sided error $[a,b]$-protocol, erring only on inputs $(x,y) \in F^{-1}(z)$, then there is a z-monochromatic rectangle of F of dimensions at least $u/2^{a+b+2} \times v/2^{b+2}$.*

In particular, any lower-bound proven using the richness method also shows a lower-bound for zero-error ("ZPP") protocols. Then our goal is to construct a problem with short zero-error protocols, but for which we can prove a large deterministic lower-bound. We start by showing the following:

Theorem 5.5. *There exists a promise problem Z, having zero-error randomized query complexity $O(1)$, but deterministic parity decision-tree complexity $\Omega(n)$.*

We may now use Theorem II (a) to lift the deterministic PDT lower-bound to the setting of asymmetric communication complexity:

Corollary 5.6. *Any deterministic $[a,b]$-protocol for $Z \circ \mathsf{MVP}_{n\times n}$ has $a = \Omega(n^2)$ or $b = \Omega(n)$, but there is a randomized, zero-error $[a,b]$-protocol for $Z \circ \mathsf{MVP}_{n\times n}$ with $a = O(n)$, $b = O(1)$.*

Since the promise problem $F = Z \circ \mathsf{MVP}_{n\times n}$ has zero-error $[O(n), O(1)]$-protocol, it then follows that the richness method cannot give a lower-bound against $[a,b]$-protocols computing F, that achieves $a = \omega(n)$. We thus established Theorem III.

Proof of Theorem 5.5. Let $Z^{-1}(0) \subseteq \{0,1\}^{2n}$ be the set of binary strings which have $z_i = 0$ whenever i is odd, and $z_i = 1$ for at least $\frac{n}{10}$-many even coordinates i. Let $Z^{-1}(1) \subseteq \{0,1\}^{2n}$ have instead $z_i = 0$ whenever i is *even*, and $z_i = 1$ for at least $\frac{n}{10}$-many *odd* coordinates i; let $Z : \{0,1\}^{2n} \to \{0,1\}$

be the corresponding promise problem. Also, let $\Delta \subseteq \{0,1\}^n$ be a binary linear code with distance $\geq \frac{n}{10}$ and constant rate $\rho = \frac{\log|\Delta|}{n} > 0$. E.g. a Justesen code [Jus72].

The upper-bound is trivial: the zero-error algorithm queries a pair $x_{2i}x_{2i+1}$; if it equals 00, the algorithm answers "I don't know", which happens with only constant probability, and otherwise the algorithm knows the answer.

The lower-bound rests on the following:

Claim 5.7. *Any vector space $V \subseteq \mathbb{F}_2^{2n}$ disjoint from $Z^{-1}(0)$ or disjoint from $Z^{-1}(1)$ must have codimension $\geq \rho \cdot n$.*

The proof of this claim is akin to the Hamming bound for codes. Let us prove the codimension lower-bound assuming V is disjoint from $Z^{-1}(0)$; the other case is proven in the same way.

Define the set $\Delta_0 \subseteq \{0,1\}^{2n}$ by placing the bits of the Δ-codewords at the even positions, and setting the odd positions to zero. For $c \in \{0,1\}^{2n}$, let $B^{(0)}(c) = c + \Delta_0$ be the set of words obtained from c by bitwise-XORing a word from Δ_0.

Suppose we had $B^{(0)}(v') \cap B^{(0)}(v'') \neq \varnothing$ for distinct $v', v'' \in V$, say $v' + \delta' = v'' + \delta''$. Then it would follow that $v \stackrel{\text{def}}{=} v' - v'' = \delta'' - \delta'$ is both in V, since $v' - v''$ is in V, and in Δ_0, since $\delta'' - \delta'$ is in Δ_0. But $\Delta_0 \subseteq Z^{-1}(0)$, since the distance of the code Δ is at least $\frac{n}{10}$. Hence, by contradiction, we must conclude that $B^{(0)}(v')$ and $B^{(0)}(v'')$ are disjoint for every distinct $v', v'' \in V$.

It then holds that $|V| \leq 2^{2n}/|\Delta_0|$ which is $\leq 2^{2n-\rho n}$ since the code Δ has rate ρ. So V has co-dimension $\geq \rho n$. This proves the claim.

Now take any deterministic parity decision-tree of depth $t < \rho \cdot n$. Consider what happens when every query $q_1, \ldots, q_t$ is answered 0. Suppose without loss of generality that the parity-decision-tree answers 1. Let $V \subseteq \{0,1\}^{2n}$ be the subspace defined by the linear equations $q_i \cdot x = 0$. Then V has co-dimension $< \rho n$, and so $V \cap Z^{-1}(0) \neq \varnothing$; but this means that the given tree does not correctly compute Z. □

5.3 Randomized simulation

The following is a restatement of Theorem II (b):

Theorem 5.8 (Randomized simulation theorem)**.** *Let n, $p \leq m = \frac{n}{1000}$ and $C < \frac{m}{100}$ be natural numbers, and let π be any randomized two-player $[Cn, C]$-protocol — where Alice gets an input from $\{0,1\}^{p\times n}$ and Bob gets an input from $\{0,1\}^n$ — computing $f \circ \mathsf{MVP}_{p\times n}$ with success probability ρ.*

Then there exists a randomized parity decision-tree which on input z computes $f(z)$ with success probability $\geq \rho - 2^{-\Omega(n)}$, after making $\leq 200 \cdot C$ parity queries.

Theorem 5.8 is a corollary of the following:

Theorem 5.9. *Let n, $p \leq m = \frac{n}{1000}$ and $C < \frac{p}{100}$ be natural numbers, and abbreviate $g = \mathsf{MVP}_{p\times n}$. Let π be any randomized two-player $[Cn, C]$-protocol where Alice gets an input from $\{0,1\}^{p\times n}$ and Bob gets an input from $\{0,1\}^n$. For a given $z \in \{0,1\}^p$, let $\pi^{-1}(z)$ denote the distribution on transcripts of π obtained by choosing a random input $(x, y) \in \{0,1\}^{p\times n} \times \{0,1\}^n$ uniformly from the inverse image $g^{-1}(z)$, and then running π on (x, y).*

Then there exists a randomized parity decision-tree which on input z outputs a string $\tau(z)$ whose distribution is 2^{-5m}-close to $\pi^{-1}(z)$ in statistical distance; this tree makes $\leq 200 \cdot C$ parity queries to z.

To see the implication, compute the probability that the label associated $\tau(z)$ is not $f(z)$:

$$\begin{aligned}\Pr_{\tau\sim\tau(z)}[\text{Label of leaf } \tau \text{ is } f(z)] &\geq \Pr_{t\sim\pi^{-1}(z)}[\text{Label of leaf } t \text{ is } f(z)] - 2^{-5m} \\ &= \Pr_{(x,y)\sim g^{-1}(z)}[\pi(x,y) = f\circ g(x,y)] - 2^{-5m} \\ &\geq \rho - 2^{-5m}\end{aligned}$$

Proof of Theorem 5.9. For the task at hand, we may assume without loss of generality that π is deterministic.[14] Here is an outline of the proof, which is similar to [GPW17]: the randomized decision-tree τ appears in Algorithm 5 below; on input z, this decision tree outputs some transcript $\tau(z)$ of a protocol $\bar{\pi}$ which is a *refinement* of π. The protocol $\bar{\pi}$ is a *refinement* in that $\bar{\pi}$ sends the same bits as π does, plus some extra bits, so that the transcript $\pi(x,y)$ always appears as a substring of the transcript $\bar{\pi}(x,y)$. Having defined $\bar{\pi}$ helps us to bound the statistical distance between $\bar{\pi}^{-1}(z)$ and $\tau(z)$ — and since $\bar{\pi}$ is a refinement of π, removing the extra bits from $\tau(z)$ will give an equally-good approximation of $\pi^{-1}(z)$.

The refined protocol $\bar{\pi}$ is given in Algorithm 4. The algorithm is written with some lines marked as 'extra bits' and it may be seen that the output of the algorithm without the extra bits is exactly the same as that of π. If we have a transcript $\bar{\pi}(x,y)$ of $\bar{\pi}$, the following claim is easy to verify:

Claim 5.10. *If we remove the extra bits from the transcript $\bar{\pi}(x,y)$, we obtain exactly the transcript $\pi(x,y)$.*

Let $\bar{\pi}^{-1}(z)$ denote the distribution on transcripts of $\bar{\pi}$ obtained by first choosing a random input $(\mathbf{x},\mathbf{y})$ uniformly from the inverse image $g^{-1}(z)$, and then running $\bar{\pi}(\mathbf{x},\mathbf{y})$. Then Claim 5.10 implies that if we sample a transcript according to the distribution $\bar{\pi}^{-1}(z)$ and remove the extra bits, the resulting string is a transcript of π distributed according to $\pi^{-1}(z)$.

The randomized decision-tree for simulating $\bar{\pi}^{-1}(z)$ appears in Algorithm 5. One may observe that it is similar to the deterministic simulation of Theorem 5.1.

More precisely, one may see by inspecting Algorithms 4 and 5 that there is an outer **while** loop which goes down through each node v of the protocol-tree of π. This is similar to the deterministic proof, however now $\tau(z)$ must find not just *one* leaf where z is represented, but rather it should produce a distribution over such leaves which is close to $\bar{\pi}^{-1}(z)$. We still want to ensure that the rectangles $A\times B$ associated with each node of the protocol tree obey certain properties — namely A should have high linear min-entropy and B should remain large.

For this purpose we associate — with each v of the outer **while** loop:

- a rectangle $A\times B\subseteq\{0,1\}^{p\times n}\times\{0,1\}^n$ which is a subrectangle of the rectangle which π associates with v;
- a set $Q\subset\{0,1\}^p$ of linearly independent vectors;
- a (possibly empty) string $w\in\{0,1\}^{|Q|}$; and
- a (possibly empty) matrix $u\in\{0,1\}^{|Q|n}$.

Then the following will be enforced:

[14] Because any randomized protocol is the convex combination of deterministic protocols, and if we can approximate the transcript of each deterministic protocol in this convex combination, when the input is drawn uniformly from $g^{-1}(z)$, then we can take the same convex combination of these approximations to obtain an approximation of the transcript of the randomized protocol.

Claim 5.11. *At line 5 of Algorithms 4 and 5, the following invariants always hold:*

(i) $A \subseteq \mathsf{Affine}^n(Q,u)$, $B = B(u,w)$, *and* $Q \cdot z = w$;;

(ii) $\mathsf{LH}_\infty(A,Q) \geq \frac{4}{5}$;

(iii) $|B| \geq 2^{\frac{9}{10}n}$; *and*

(iv) $|Q| \leq 200C$.

Invariant (i) follows by properties (c2) and (c3) of Lemma 4.8. Our setting $A = A'$ in line 14 preserves invariant (ii), because of Lemma 3.8. Setting $A = A_i$ in line 28 also preserves invariant (ii), because of property (c4) of Lemma 4.8. Finally, invariants (iii) and (iv) are enforced at the end of each cycle, in lines 30-35.

Sometimes the refined protocol $\bar{\pi}$ will fail to enforce invariants (i-iv). This can happen in lines 11, 21, 25, 31, or 34, and when it does happen, one of the players will send the *abort symbol* $\perp$ to the other player, and from then onward both players run protocol π starting from node v in π's protocol tree. Let us say that $\bar{\pi}^{-1}(z)$ *aborts* when this does happen, i.e. a transcript sampled according to $\bar{\pi}^{-1}(z)$ aborts if it contains the abort symbol $\perp$.

We now show that the procedure correctly approximates $\bar{\pi}^{-1}(z)$ on input z:

Claim 5.12. *For every $z \in \{0,1\}^p$, the two distributions $\bar{\pi}^{-1}(z)$ and $\tau(z)$ are 2^{-5m}-close in statistical distance.*

We will show this in three steps. Let $\tilde{\pi}^{-1}(z)$ equal $\bar{\pi}^{-1}(z)$ conditioned on not aborting, and let $\tilde{\tau}(z)$ equal $\tau(z)$ conditioned on non-aborting. We first show $\bar{\pi}^{-1}(z)$ is statistically close to $\tilde{\pi}^{-1}(z)$, which in turn is close to $\tilde{\tau}(z)$, which in turn is close to $\tau(z)$.

To show the first, we upper-bound the probability that $\bar{\pi}$ aborts. Invariants (i-iii) hold at each line where the algorithm may abort (lines 11, 21, 25, 31, or 34). Let $(\mathbf{x},\mathbf{y})$ be chosen uniformly at random from the set $A \times B \cap g^{-1}(z)$, at any of these lines. It then follows, from the inverse-marginals lemma (Lemma 4.9) that the marginal distribution of $\mathbf{x}$ is 2^{-8m}-close to uniform on A, and the marginal distribution of $\mathbf{y}$ is 2^{-8m}-close to uniform on B. With this in mind and noting that we have initialized $\varepsilon = 2^{-10m}$, we may now bound the probability of aborting:

At line 11 we have $\Pr[\mathbf{x} \in A \setminus A'] \leq \varepsilon + 2^{-8m} \ll 2^{-5m}$;

At line 21 we have $\Pr[\mathbf{x} \in A^\dagger] \leq \varepsilon + 2^{-8m} \ll 2^{-5m}$;

At line 25 we have $\Pr[\mathbf{y} \in B^\dagger] \leq \varepsilon + 2^{-8m} \ll 2^{-5m}$.

At line 31 we must notice that B decreases in size at two places in the algorithm: lines 17 and 28. By property (c3) of the entropy-restoring partition (Lemma 4.8), in line 28 B will be cutoff by a total fraction which is no smaller than $(1-\varepsilon)2^{-|Q|} \geq 2^{-20\ell} \gg 2^{-\frac{n}{20}}$. On the other hand, the probability that $|B|$ is reduced by a factor γ in line 17 is no greater than $\gamma 2^{6C} + C2^{-8m}$ — since it would be no greater than $\gamma 2^{6C}$ if $\mathbf{y}$ would be uniform on B (by Lemma 3.13), $\mathbf{y}$ is 2^{-8m}-close to uniform, and line 17 is executed C times (once for each bit Bob communicates in π). Hence $\gamma < 2^{-\frac{n}{20}}$ with probability at most $2^{-\frac{n}{20}+6C} + C2^{-8m} \ll 2^{-5m}$, and otherwise $|B|$ will remain at least $2^{-\frac{n}{10}}2^n$.

At line 34 we must notice the following: whenever Q gains k new elements — which only happens in line 29 — property (c4) of Lemma 4.8 says that the *affine-density* $\frac{|A|}{|\mathsf{Affine}^n(Q,u)|}$ will increase by $2^{\frac{1}{20}kn}$. Now, this affine density can only decrease in line 14, and as above the probability

that this affine density is reduced by a factor γ is no greater than $\gamma \cdot 2^{6Cn} + Cn \cdot 2^{-8m}$. (The reason is similar to that in the previous paragraph — Alice's communication is Cn, and hence the first term comes form Lemma 3.13 assuming $\mathbf{x}$ is uniform on A, and the second term is from the fact that $\mathbf{x}$ is 2^{-8m} close to uniform.) For $\gamma \leq 2^{-10Cn}$, the probability that the affine-density of A decreases by a factor smaller than 2^{-10Cn} is $\leq \gamma \cdot 2^{6Cn} + Cn \cdot 2^{-8m} \ll 2^{-5m}$. And so, since the affine density is never greater than 1, it follows that $|Q|$ will remain bounded by $200 \cdot C$ with that much probability.

It then follows that $\bar{\pi}^{-1}(z)$ aborts with probability $\ll 2^{-5m}$, and hence the statistical distance between $\tilde{\pi}^{-1}(z)$ and $\bar{\pi}^{-1}(z)$ is $\ll 2^{-5m}$. It is easy to see that the statistical distance between $\tilde{\pi}^{-1}(z)$ and $\tilde{\tau}(z)$ is also small. The two processes behave the same, except in the way that A and B are updated; for example in line 13 of Algorithm 4, in order to choose $i \in \{0,1\}$, $\tilde{\pi}^{-1}(z)$ will choose uniformly-random pair $(\mathbf{x}, \mathbf{y})$ in $g^{-1}(z) \cap A' \times B$, and then set $i = 0$ if $\mathbf{x} \in A' \cap A_{v_0}$, and set $i = 1$ if $\mathbf{x} \in A' \cap A_{v_1}$; $\tilde{\tau}(z)$ will do the same in line 11 of Algorithm 5, except $\mathbf{x}$ is uniformly chosen from A' instead. But by the inverse-marginals lemma (Lemma 4.9), the two resulting distributions on i are 2^{-8m}-close. The same happens at every other point when the transcript is updated in both Algorithms 4 and 5. Since the length of the transcript is $O(Cn) \ll 2^p$, it follows that the total statistical distance between $\tilde{\pi}^{-1}(z)$ and $\tilde{\tau}(z)$ is $\ll 2^{-5m}$. The proof that $\tilde{\tau}(z)$ and $\tau(z)$ are close is almost identical to the proof that $\tilde{\tau}^{-1}(z)$ and $\tau^{-1}(z)$ are close, except that it no longer requires the use of the inverse-marginals lemma.

This concludes the proof of Claim 5.12. We are left only to observe that the set Q contains the parity queries made to z, and $|Q|$ is forcefully bounded in lines 28–29 of Algorithm 5. So when we bounded the probability of the corresponding abort condition we also bounded the number of parity queries made by Algorithm 5. □

Algorithm 4 Refined **deterministic** communication protocol $\bar{\pi}$

input $x \in \{0,1\}^{p \times n}$ (to Alice) and $y \in \{0,1\}^n$ (to Bob).
Both players initialize $A = \{0,1\}^{p \times n}$, $B = \{0,1\}^n$.
Both players initialize $\varepsilon = 2^{-n/100}$, $Q = \varnothing$, $w = \varnothing$, $u = \varnothing$.
Let v be the root of π.
while v is not a leaf **do**

Invariants which are true here are discussed in page 29.

  if $\mathsf{LH}^{\varepsilon}_{\infty}(A, Q) \geq \frac{9}{10}$ **then**
    Let v_0 and v_1 be the children of v.
    if Alice communicates at v **then**
      Let $A' \subseteq A$ have $\mathsf{LH}_{\infty}(A', Q) \geq \frac{9}{10}$ and $|A'| \geq (1-\varepsilon)|A|$.[a]
      if $x \in A \setminus A'$ **then**
        Alice sends $\perp$, ▷ Extra bits
        Both players **run** $\pi_v(x, y)$ and **exit**.
      Alice sends i such that $x \in A_{v_i}$.
      Both players update $A = A' \cap A_{v_i}$ and $v = v_i$.
    else if Bob communicates at v **then**
      Bob sends i such that $y \in B_{v_i}$.
      Both players update $B = B_{v_i}$ and $v = v_i$.
  else if $\mathsf{LH}^{\varepsilon}_{\infty}(A, Q) < \frac{9}{10}$ **then**
    Apply Lemma 4.8 with $N = 1$ and $B_1 = B$, to obtain
    - Partitions $A = A^{\dagger} \cup A_1 \cup A_2 \cdots$ and $B = B^{\dagger} \cup B'$;
    - For each $i \geq 1$, the values of k'_i, Q'_i, and u'_i.

    if $x \in A^{\dagger}$ **then**
      Alice sends $\perp$, ▷ Extra bits
      Both players **run** $\pi_v(x, y)$ and **exit**.
    Alice sends i to Bob such that $x \in A_i$.
    if $y \in B^{\dagger}$ **then**
      Bob sends $\perp$, ▷ Extra bits
      Both players **run** $\pi_v(x, y)$ and **exit**.
    Bob sends $w'_i = u'_i \cdot y$ to Alice.
    Both players update $A = A_i$, $B = B'(u'_i, w'_i)$;
    Both players update $Q = Q \cup Q'_i$, $u = uu'_i$ and $w = ww'_i$.
  if $|B| < 2^{\frac{9}{10}n}$ **then**
    Bob sends $\perp$, ▷ Extra bits
    Both players **run** $\pi_v(x, y)$ and **exit**.
  if $|Q| > 200C$ **then**
    Alice sends $\perp$, ▷ Extra bits
    Both players **run** $\pi_v(x, y)$ and **exit**.
Output the label of the leaf v.

[a] Such A' exists since A has large smooth linear min-entropy; more than one such A' may exist, any choice is valid.

Algorithm 5 Randomized decision-tree procedure $\tau(z)$ simulating $\bar{\pi}^{-1}(z)$

input $z \in \{0,1\}^p$.
Initialize $A = \{0,1\}^{p \times n}$, $B = \{0,1\}^n$.
Initialize $\varepsilon = 2^{-n/100}$, $Q = \varnothing$, $w = \varnothing$, $u = \varnothing$.
Let v be the root of π and $\tau = \varnothing$.
while v is not a leaf **do**

Invariants (i-iv) also hold here.

  if $\mathsf{LH}^{\varepsilon}_{\infty}(A, Q) \geq \frac{9}{10}$ **then**
    Let v_0 and v_1 be the children of v.
    if Alice communicates at v **then**
      Let $A' \subseteq A$ have $\mathsf{LH}_{\infty}(A', Q) \geq \frac{9}{10}$ and $|A'| \geq (1-\varepsilon)|A|$.
      With probability $\frac{|A \setminus A'|}{|A|}$, **output** τ and **abort**.
      Choose $i \in \{0,1\}$ with probability $\frac{|A' \cap A_{v_i}|}{|A'|}$ and set $\tau = \tau i$.
      Update $A = A' \cap A_{v_i}$ and $v = v_i$.
    else if Bob communicates at v **then**
      Choose $i \in \{0,1\}$ with probability $\frac{|B \cap B_{v_i}|}{|B|}$ and set $\tau = \tau i$.
      Update $B = B_{v_i}$ and $v = v_i$.
  else if $\mathsf{LH}^{\varepsilon}_{\infty}(A, Q) < \frac{9}{10}$ **then**
    Apply Lemma 4.8 with $N = 1$ and $B_1 = B$, to obtain
    - Partitions $A = A^{\dagger} \cup A_1 \cup A_2 \cdots$ and $B = B^{\dagger} \cup B'$;
    - For each $i \geq 1$, the values of k'_i, Q'_i, and u'_i.

    With probability $\frac{|A^{\dagger}|}{|A|}$, **output** τ and **abort**.
    Choose $i \geq 1$ with probability $\frac{|A_i|}{|A \setminus A^{\dagger}|}$
    Set $\tau = \tau\ i$.
    With probability $\frac{|B^{\dagger}|}{|B|}$, **output** τ and **abort**.
    Query z to discover $w'_i = u'_i \cdot z$.
    Set $\tau = \tau\ w'_i$.
    Update $A = A_i$, $B = B(u'_i, w'_i)$;
    Update $Q = Q \cup Q'_i$, $u = uu'_i$ and $w = ww'_i$.
  if $|B| < 2^{\frac{9}{10}n}$ **then**
    Output τ and **abort**.
  if $|Q| > 200C$ **then**
    Output τ and **abort**.
Output τ.

5.4 Counting orthogonal vectors

Definition 5.13. The problem of *counting orthogonal vectors*, $\mathsf{OVC}_n : \{0,1\}^{n\times n} \times \{0,1\}^n \to \{0,\ldots,n\}$, is the static data-structure problem of encoding a set $x \subseteq \{0,1\}^n$ of n-many n-bit vectors, so that we may know, for any given query $y \in \{0,1\}^n$, how many x_i are orthogonal to y:

$$\mathsf{OVC}_n(x,y) = |\{i \in [n] \mid \mathsf{IP}_n(x_i,y) = 0\}|.$$

The problem of *counting orthogonal vectors mod-3*, $\mathsf{OVC}^3_n : \{0,1\}^{n\times n} \times \{0,1\}^n \to \{0,1\}$, is the static data-structure problem of encoding a set $x \subseteq \{0,1\}^n$ of n-many n-bit vectors, so that we may know, for any given query $y \in \{0,1\}^n$, if the number of x_i orthogonal to y is a multiple of 3:

$$\mathsf{OVC}_n(x,y) = \begin{cases} 1 & \text{if } \mathsf{OVC}_n(x,y) = 0 \mod 3, \\ 0 & \text{otherwise.} \end{cases}$$

The *span* problem, $\mathsf{Span}_n : \{0,1\}^{n\times n} \times \{0,1\}^n \to \{0,1\}$ is the static data-structure problem of encoding a set $x \subseteq \{0,1\}^n$ of n-many n-bit vectors, so that we may know, for any given query $y \in \{0,1\}^n$, whether all x_i are orthogonal to y:

$$\mathsf{Span}_n(x,y) = \begin{cases} 1 & \text{if } \mathsf{OVC}_n(x,y) = n, \\ 0 & \text{otherwise.} \end{cases}$$

The *orthogonal majority* problem, $\mathsf{OGMaj}_n : \{0,1\}^{n\times n} \times \{0,1\}^n \to \{0,1\}$ is the static data-structure problem of encoding a set $x \subseteq \{0,1\}^n$ of n-many n-bit vectors, so that we may know, for any given query $y \in \{0,1\}^n$, whether most x_i are orthogonal to y:

$$\mathsf{OGMaj}_n(x,y) = \begin{cases} 1 & \text{if } \mathsf{OVC}_n(x,y) \geq \frac{n}{2}, \\ 0 & \text{otherwise.} \end{cases}$$

The following theorem has been shown in [MNSW98]:[15]

Theorem 5.14 ([MNSW98])**.** *There exists a real constant $\varepsilon > 0$ such that there are no deterministic $[S,w,t]$-schemes for Span_n with $w \leq n$, unless when $t \geq \frac{\varepsilon n}{\log \frac{Sw}{n}}$.*

By simple reductions, the above lower-bound also holds against OVC_n and OGMaj_n. On the other hand, it is easy to show the following upper-bound for randomized schemes:

Theorem 5.15. *There is a randomized $[O(n^2), n, O(1), \Omega(1)]$-scheme for Span_n.*

Proof. Our encoding will contain S vectors $q_1 \cdot x, \ldots, q_S \cdot x$, where each $q_i \in \{0,1\}^n$. I.e., each $q_i \cdot x$ is the bitwise XOR of some rows of x.

Suppose we pick a vector $q \in \{0,1\}^n$ uniformly at random. Then any such choice will always have $q \cdot x \cdot y = 0$ if $x \cdot y = 0$; but if $x \cdot y \neq 0$, then $\Pr_q[q \cdot x \cdot y = 0] = \frac{1}{2}$. So if we pick $S = O(n)$ vectors $q_1, \ldots, q_s$, it is overwhelmingly likely (by, e.g., a Chernoff bound) that for every $y \in \{0,1\}^n$ with $x \cdot y \neq 0$, roughly half of the q_i will have $q_i \cdot x \cdot y = 1$. Our encoding $E(x)$ is $q_1 \cdot x, \ldots, q_S \cdot x$ for one of these overwhelmingly likely choices for $q_1, \ldots, q_S$. The query algorithm will now pick $O(1)$ random $i \in S$ and obtain the vectors $q_i \cdot x$. If all $q_i \cdot x \cdot y$ equal 0, then the algorithm concludes, with only constant error probability, that $\mathsf{Span}_n(x,y) = 1$; if some $q_i \cdot x \cdot y$ equals 1, then the algorithm concludes, correctly, that $\mathsf{Span}_n(x,y) = 0$. □

[15] The proof is via their *richness technique*, which we briefly overview in the next section.

The same idea as above will give us a randomized $[n, O(1), \Omega(1)]$-protocol for Span_n. Hence there is no hope to show a randomized lower-bound for Span_n. We are, however, able to show the following theorem, which is a restatement of Theorem I (b) and (c):

Theorem 5.16. *There exists a real constant $\varepsilon > 0$ such that:*

- *In any randomized $[s, w, t, \rho]$-scheme for OVC^3_n with $w \leq n$, either $t \geq \frac{\varepsilon n}{\log \frac{sw}{n}}$ or $\rho \leq \frac{2}{3} + 2^{-\varepsilon n}$.*
- *In any randomized $[s, w, t, \rho]$-scheme for OGMaj_n with $w \leq n$, either $t \geq \frac{\varepsilon n}{\log \frac{sw}{n}}$ or $\rho \leq 1 - \varepsilon$.*

Remarkably, the contrast between these two situations can be explained by the fact that the negated OR function has small randomized PDT complexity, whereas the counting and majority functions have large randomized PDT complexity. We will now prove these PDT lower-bounds, and Theorem 5.16 will then follow directly from our randomized simulation theorem (Theorem 5.8) and from the known connection between communication complexity and data structures (Corollary 3.12).

Theorem 5.17. *Any randomized parity decision tree complexity needs $\Omega(n)$ queries to compute $\mathsf{Mod3}^0_n$ with success probability $> \frac{2}{3} + 2^{-\Omega(n)}$.*

Proof. We show that any deterministic parity decision tree which solves $\mathsf{Mod3}_n$ on the uniform distribution over $\{0,1\}^n$ must make $\Omega(n)$ queries in order for the error probability to be bounded away from $\frac{1}{2}$. Consider any such deterministic parity decision tree, and let us assume that the number of queries it makes in the worst case is $\ell = o(n)$. Let us also denote these queries as $Q = (q_1, \cdots, q_\ell)$, and their respective answers as $b = (b_1, \cdots, b_\ell)$. The tuple (Q, b) specify an affine subspace $V(Q, b)$ of codimension ℓ in a natural way. Next we show that in any affine subspace V of codimension $o(n)$, and for any $a \in \{0, 1, 2\}$, the fraction of $x \in V$ such that $\sum_{i=1}^n x_i = a \mod 3$, is in the range $\frac{1}{3} \pm o(1)$.

Consider the single affine constraint $q \cdot x = b$, given by $q \in \{0,1\}^n$ and $b \in \{0,1\}$. Note that $q \cdot x$ is an n variate multi-linear degree 1 polynomial over $\mathbb{Z}_2$. If $f : \{0,1\}^n \to \mathbb{Z}_3$, then let by $\mathsf{PDisc}^{q,b}(f)$ be given by

$$\mathsf{PDisc}^{q,b}(f) = \max_{a \in \mathbb{Z}_3} \left| \Pr_x[f(x) \wedge q \cdot x = b] - \frac{1}{3} \Pr_x[q \cdot x = b] \right|$$

We use the following special case of the *polynomial discrepancy lemma*, appearing in [Cha07]:

Lemma 5.18 (Polynomial discrepancy lemma [Cha07])**.** *There exists a constant β such that, for any $q \in \{0,1\}^n$ and any $b \in \{0,1\}$:*

$$\mathsf{PDisc}^{q,b}(\mathsf{Mod3}_n) \leq \exp(-\beta n).$$

Let $V = \{x \in \mathbb{F}_2^n \mid \forall i \in [\ell] q_i \cdot x = b_i\}$ be an arbitrary affine subspace of codimension ℓ, given by the linear constraints $q_i \in \{0,1\}^n$, and $b_i \in \{0,1\}^\ell$; let $\mathbf{E}_x$ denote expectation over a uniformly random $x \in \{0,1\}^n$; if $\varnothing \neq S \subseteq [\ell]$, let $q_S = \bigoplus_{i \in S} q_i$ (the bitwise-XOR of q_i with $i \in S$) and $b_S = \bigoplus_{i \in S} b_i$

(the XOR of the b_i with $i \in S$). We may then write

$$\begin{aligned}
\Pr_{x\in\{0,1\}^n}[\mathsf{Mod3}_n^a(x) \wedge x \in V] &= \mathbf{E}_x\left[\mathsf{Mod3}_n^a(x)\prod_{j\in[\ell]}\left(\frac{1+(-1)^{q_j\cdot x-b_j}}{2}\right)\right] \\
&= 2^{-\ell}\cdot\mathbf{E}_x\left[\mathsf{Mod3}_n^a(x)\left(1+\sum_{\varnothing\neq S\subseteq[\ell]}(-1)^{q_S\cdot x-b_S}\right)\right] && \text{(expanding the product)} \\
&\in 2^{-\ell}\left(\frac{1}{3}\pm e^{-\beta n}\right)+2^{-\ell}\sum_{\varnothing\neq S\subseteq[\ell]}\mathbf{E}_x\left[\mathsf{Mod3}_n^a(x)\cdot(-1)^{q_S\cdot x-b_S}\right] && \text{(Lemma 5.18)} \\
&= 2^{-\ell}\left(\frac{1}{3}\pm e^{-\beta n}\right)+2^{-\ell}\sum_{\varnothing\neq S\subseteq[\ell]}\sigma_S,
\end{aligned}$$

where

$$\begin{aligned}
\sigma_S &= \mathbf{E}_x\left[\mathsf{Mod3}_n^a(x)\cdot(-1)^{q_S\cdot x-b_S}\right] \\
&= \frac{1}{2^n}\sum_{x\equiv a \mod 3}(-1)^{q_S\cdot x-b_S} \\
&= \frac{1}{2^n}\left(\sum_{x\equiv a \mod 3}[q_S\cdot x=b_S]-\sum_{x\equiv a \mod 3}[q_S\cdot x=1-b_S]\right) \\
&\in \frac{1}{2^n}\left(2^{n-1}\left(\frac{1}{3}\pm e^{-\beta n}\right)-2^{n-1}\left(\frac{1}{3}\pm e^{-\beta n}\right)\right) && \text{(Lemma 5.18)} \\
&\subseteq \pm e^{-\beta n}
\end{aligned}$$

This then implies that

$$\Pr_x[\mathsf{Mod3}_n(x)=a\wedge Q\cdot x=b]\in 2^{-\ell}\left(\frac{1}{3}\pm 2\cdot e^{-\beta n}\right)=\Pr_x[Q\cdot x=b]\left(\frac{1}{3}\pm 2\cdot e^{-\beta n}\right),$$

i.e., in an affine subspace of codimention ℓ, the fraction of elements x such that $\mathsf{Mod3}_n(x)=a$ for any $a\in\{0,1,2\}$ is $\frac{1}{3}\pm 2^{-cn}$ for some constant c. This, in turn, means that in each leaf of a parity decision-tree for $\mathsf{Mod3}_n^0$, the decision tree makes an error of at least $1/3-2^{-\Omega(n)}$ for any $\ell=o(n)$. □

By a binary-search reduction to majority, we may easily see that the randomized parity decision tree complexity of GMaj_n is $\Omega\left(\frac{n}{\log n\log\log n}\right)$. However the following stronger result is known:

Theorem 5.19 ([CR12, Vid12, She12])**.** *The randomized pdt complexity of* GMaj *is* $\Omega(n)$.

6 Lower-bounds for the VMV problem

Consider the communication problem where Alice gets a $p\times n$-bit matrix x, Bob gets a p-bit vector q and an n-bit vector y, and they wish to compute the vector-matrix-vector product $\mathsf{VMV}_{p\times n}(x;q,y)=q\cdot x\cdot y$.

This problem intuitively feels similar to the $f\circ\mathsf{MVP}_{p\times n}$ problem treated in the simulation theorems of Section 5. However, instead of computing an outer function f of the matrix-vector

product $x \cdot y$, Bob has a certain parity-query q and they wish to know $q \cdot (x \cdot y)$.[16] So let us suppose we have a communication protocol for the $\mathsf{VMV}_{p\times n}$ problem... does that correspond, via a simulation-type argument, to a decision-tree of some kind?

We show that this is exactly the case, that an efficient protocol for the VMV problem would give us an efficient decision tree for a certain *impossible task*:

Lemma 6.1 (Impossible task)**.** *Suppose we have a randomized parity decision-tree running in time t which, on every input $z \in \{0,1\}^p$, outputs a pair $(q,b) \in \{0,1\}^p \times \{0,1\}$ such that both:*

- *q is (always) linearly-independent of the set Q of parity queries that were made, and*
- *with probability ρ over the choice of q, we have $q \cdot z = b$.*

Then either $t \geq p$ or $\rho = 1/2$.

Proof. Suppose without loss of generality that the given parity decision-tree is a convex combination of deterministic parity decision-trees which do exactly t linearly-independent queries at each path. This can be ensured by padding the set of queries if they are fewer than t, and by removing redundant linearly-dependent queries.

Suppose τ is a deterministic parity decision-tree in this convex combination, and suppose that $t < p$. We know that:

(a) on every input $z \in \{0,1\}^p$, τ makes exactly t linearly-independent parity-queries $Q = Q(z)$ and outputs a pair $(q,b) \in \{0,1\}^p \times \{0,1\}$ where q is linearly independent of Q; and

(b) for some ρ-fraction of the possible inputs, the output pair (q,b) is such that $q \cdot z = b$.

We may now look at a arbitrary leaf of the tree τ, and consider the set $Z \subseteq \{0,1\}^p$ of inputs which end up in this leaf. Notice that $|Z| = 2^{p-t} \geq 2$, and hence, no matter which pair (q,b) is chosen as the output for the given leaf, $b = q \cdot z$ will hold for exactly $\frac{|Z|}{2}$ many $z \in Z$. Since this happens for every leaf of τ, it follows that $\rho = \frac{1}{2}$ exactly. As τ was an arbitrary decision-tree in the convex combination of the given randomized decision-tree, it will follow that the probability $\rho = \frac{1}{2}$ for this convex combination, also. □

6.1 Statement of the communication-complexity lower-bound

We now present an analogue of Theorem 5.8. The proof is very similar, and will be sketched in Section 6.3.

Theorem 6.2. *Let n, $p \leq m = \frac{n}{1000}$ and $C < \frac{p}{4\times 10^5}$ be natural numbers, and abbreviate $g = \mathsf{MVP}_{p\times n}$. Let π be any randomized two-player $[Cn, C]$-protocol where Alice gets an input $x \in \{0,1\}^{p\times n}$ and Bob gets an input $(q,y) \in \{0,1\}^p \times \{0,1\}^n$, and which outputs $q \cdot x \cdot y$ with probability ρ_0.*

Then there exists a randomized parity decision-tree for solving the impossible task of Lemma 6.1, with success probability $\rho \in \rho_0 \pm 2^{-\Omega(p)}$

It then follows from Lemma 6.1 that either $C = \Omega(p)$, or $\rho = \frac{1}{2}$, and so we get:

Corollary 6.3. *There exists some constant $\varepsilon > 0$ for which the following holds: In any randomized $[a,b]$-protocol for solving the $\mathsf{VMV}_{n\times n}$ problem with success probability ρ, we must have $a \geq \varepsilon \cdot n^2$, or $b \geq \varepsilon \cdot n$, or $\rho < \frac{1}{2} + 2^{-\varepsilon \cdot n}$.*

[16] Interestingly, if Alice knew what q is, she could just send $q \cdot x \in \{0,1\}^n$ to Bob and then Bob would immediately know the answer. Any lower-bound we purport to prove must then use the fact that Bob's communication budget is too small for Bob to send q to Alice.

6.2 The data-structure lower-bound

Now consider the VMV static data-structure problem: We wish to encode an $n \times n$ matrix $x \in \{0,1\}^{n \times n}$ and be able to compute $q \cdot x \cdot y$, for any given query $(q, y) \in \{0,1\}^n \times \{0,1\}^n$. Then Corollary 3.12 and Corollary 6.3 give us the following restatement of Theorem I (a):

Theorem 6.4. *There exists some constant $\varepsilon > 0$ such that any $[s, w, t, \rho]$-scheme for the $\mathsf{VMV}_{n \times n}$ problem, with $w \leq n$, has either $t \geq \frac{\varepsilon n}{\log \frac{sw}{n}}$ or $\rho \leq \frac{1}{2} + 2^{-\varepsilon n}$.*

The four-Russians algorithm [ADKF70, Wil07] shows that the above lower-bound is optimal for large w:

Theorem 6.5. *There exists an $[\frac{n^2}{\log n}, n, \frac{n}{\log n}, 1]$-scheme for the VMV problem.*

Proof. To encode x, partition the rows of x into $t = \frac{n}{\log n}$ blocks $x_1, \ldots, x_t$, each having $\log n$ rows; then store, for each block x_i of rows and for each $q' \in \{0,1\}^{\log n}$, the single row $q' \cdot x_i$ in a single cell of size $w = n$. This uses $S = \frac{n^2}{\log n}$ cells.

Given a query q, y, the deterministic query algorithm breaks q into t blocks $q_1, \ldots, q_t$, and obtains $q_i \cdot x_i$ for each $i \in [t]$, thus making exactly t queries; then the output is the XOR: $q \cdot x \cdot y = \oplus_{i \in [t]} q_i \cdot x_i \cdot y$. □

6.3 Proof of the communication complexity lower-bound

Proof of Theorem 6.2. Suppose for the time being that π is deterministic. For a given $z \in \{0,1\}^p \times \{0,1\}^p$, let $\pi^{-1}(z)$ denote the distribution on pairs (q, t), where q is a uniformly random $q \in \{0,1\}^p$, and t is a transcript of of π obtained by choosing a random input $(x, y) \in \{0,1\}^{p \times n} \times \{0,1\}^n$ uniformly from the inverse image $g^{-1}(z)$, and then running π on $(x; q, y)$.

We will construct a randomized parity decision-tree τ which on input z outputs a pair $\tau(z) = (q, t)$ whose distribution is $2^{-\Omega(p)}$-close to $\pi^{-1}(z)$ in statistical distance, and such that q is always independent of the queries made by $\tau(z)$; this tree will make $\leq 200 \cdot C$ parity queries to z.

The conclusion of the theorem will then follow: if we run $\tau(z) = (q, t)$ and output the pair (q, b), where q is the label of the transcript t (or abort if τ aborts), then q will be independent of the queries made with probability $1 - 2^{-\Omega(p)}$ (since $\leq 200C \ll p$ queries were made), and b will equal $q \cdot x \cdot y = q \cdot z$ with probability $\rho_0 \pm 2^{-\Omega(n)}$, so we will succeed at the impossible task with probability $\rho = \rho_0 - 2^{-\Omega(p)}$; then either $C \geq \frac{p}{200}$ or $\rho = \frac{1}{2}$.

If we assume that π is not deterministic, i.e., that it is a convex combination of deterministic protocols, then taking the same convex combination of the corresponding τ will give us the same conclusion.

The construction of τ is very similar to the one appearing in the proof of Theorem 5.9. The one difference is the following: instead of keeping track of a single set $B \subseteq \{0,1\}^n$, now Bob's set B is a subset of $\{0,1\}^p \times \{0,1\}^n$, which we think of as a collection of sets $B_1, \ldots, B_N$; each B_i corresponds to some string q, so that B_i is the set of *extensions* $y \in \{0,1\}^n$ such that $(q, y) \in B$. So instead of preserving the invariant that B is large, we now wish to preserve the invariant that all of the B_i sets are large.

We need to spell out these invariants carefully. For this purpose, let us use the following notation, for a given set $B \subseteq \{0,1\}^p \times \{0,1\}^n$:

- $B^{(0)} = \{q \in \{0,1\}^p \mid (q, y) \in B \text{ for some } y\}$ is the projection of B onto the first coordinate.
- For each $q \in B^{(0)}$, $B_q = \{b \mid (q, b) \in B\} \subseteq \{0,1\}^n$.

- If $u \in \{0,1\}^{k \times n}$ and $w \in \{0,1\}^k$, then $B(u,w) = \{(q,y) \in B \mid u \cdot y = w\}$.

One may see by inspecting Algorithms 6 and 7 that τ has an outer **while**-loop which goes down through each node v of the protocol-tree of π. With each v of this outer loop, we associate:

- a rectangle $A \times B \subseteq \{0,1\}^{p \times n} \times (\{0,1\}^p \times \{0,1\}^n)$ which is a subrectangle of the rectangle which π associates with v;
- a (possibly empty) set $Q \subset \{0,1\}^p$ of linearly independent vectors;
- a (possibly empty) string $w \in \{0,1\}^{|Q|}$; and
- a (possibly empty) matrix $u \in \{0,1\}^{|Q| n}$.

Then we will enforce:

Claim 6.6. *At line 5 of Algorithms 6 and 7, the following invariants always hold:*

(i) $A \subseteq \mathsf{Affine}^n(Q,u)$, $B = B(u,w)$, *and* $Q \cdot z = w$;;

(ii) $\mathsf{LH}_\infty(A,Q) \geq \frac{4}{5}$;

(iii) $|B^{(0)}| \geq 2^{\frac{9}{10}p}$ *and for every* $q \in B^{(0)}$, $|B_q| \geq 2^{\frac{9}{10}n}$; *and*

(iv) $|Q| \leq 200 \cdot C$.

Algorithms 6 and 7 are similar to Algorithms 4 and 5, respectively, and the proof that $\bar{\pi}^{-1}(z)$ is close to $\tau(z)$ is almost identical. With completeness in mind, we will still write down the whole proof, but let us here pinpoint the crucial difference: when upper-bounding the probability of aborting at lines 30-36, here instead of bounding the probability that a single B should not become too small, we must bound the probability that this happens for many B_q. For this purpose, we may argue as in item "At line 31" (of page 29), using the coin-tossing lemma (Lemma 3.13), that $|B| \geq 2^{-\frac{p}{10}+1} \cdot 2^{p+n}$ except with abort probability $\ll 2^{-5p}$ (since Bob only communicates $C \leq \frac{p}{100}$ bits), and so there must be at least $2^{\frac{9}{10}p}$-many q such that $|B_q| \geq 2^{\frac{9}{10}n}$ throughout.

Let us start by showing Claim 6.6 for Algorithm 6. Invariant (i) follows by properties (c2) and (c3) of Lemma 4.8. Our setting $A = A'$ in line 14 preserves invariant (ii), because of Lemma 3.8. Setting $A = A_i$ in line 28 also preserves invariant (ii), because of property (c4) of Lemma 4.8. Finally, invariants (iii) and (iv) are enforced at the end of each cycle, in lines 30-36. The proof is identical for Algorithm 7.

If we have a transcript $\bar{\pi}(x,y)$ of $\bar{\pi}$, the following claim is easy to verify:

Claim 6.7. *Removing the extra bits from the transcript* $\bar{\pi}(x,y)$ *gives us exactly* $\pi(x,y)$.

Let $\bar{\pi}^{-1}(z)$ denote the distribution on pairs (q,t) obtained by first choosing a random triple $(x;q,y)$, where q is uniformly chosen from $\{0,1\}^p$, (x,y) is uniformly chosen from the inverse image $g^{-1}(z)$, and $t = \bar{\pi}(x;q,y)$. Then Claim 6.7 implies that if we sample a pair (q,t) according to the distribution $\bar{\pi}^{-1}(z)$ and remove the extra bits from t, the resulting pair will be distributed according to $\pi^{-1}(z)$.

Sometimes the refined protocol $\bar{\pi}$ will fail to enforce invariants (i-iv). This can happen in lines 11, 21, 25, 32, or 35, and when it does happen, one of the players will send the *abort symbol* $\perp$ to the other player, and from then onward both players run protocol π starting from node v in π's protocol tree. Let us say that $\bar{\pi}^{-1}(z)$ *aborts* when this does happen, i.e. a transcript sampled according to $\bar{\pi}^{-1}(z)$ aborts if it contains the abort symbol $\perp$.

Now that we have explained the refined protocol $\bar{\pi}$, we are ready to present the parity decision-tree procedure which simulates it. It appears as Algorithm 7 below.

We now establish that $\tau(z)$ correctly approximates $\bar{\pi}^{-1}(z)$:

Claim 6.8. *For every $z \in \{0,1\}^p$, the two distributions $\bar{\pi}^{-1}(z)$ and $\tau(z)$ are $2^{-\frac{p}{100}}$-close in statistical distance.*

We will show this in three steps. Let $\tilde{\pi}^{-1}(z)$ equal $\bar{\pi}^{-1}(z)$ conditioned on not aborting, and let $\tilde{\tau}(z)$ equal $\tau(z)$ conditioned on non-aborting. We first show $\bar{\pi}^{-1}(z)$ is statistically close to $\tilde{\pi}^{-1}(z)$, which in turn is close to $\tilde{\tau}(z)$, which in turn is close to $\tau(z)$.

To show the first, we upper-bound the probability that $\bar{\pi}$ aborts. Invariants (i-iii) hold at each line where the algorithm may abort (lines 11, 21, 25, 32, or 35). Let $(\mathbf{x}_q, \mathbf{y}_q)$ be chosen uniformly at random from the set $A \times B_q \cap g^{-1}(z)$, at any of these lines. It then follows, from the inverse-marginals lemma (Lemma 4.9) that the marginal distribution of $\mathbf{x}_q$ is 2^{-8m}-close to uniform on A, and the marginal distribution of $\mathbf{y}_q$ is 2^{-8m}-close to uniform on B_q. Hence if we let $(\mathbf{x}, \mathbf{y})$ be chosen uniformly from the set $\{(x,y) \in A \times B_q \mid q \in Q^{(0)}, x \cdot y = z\}$, it will then hold that $\mathbf{x}$ is 2^{-8m}-close to uniform on A, and $\mathbf{y}$ is 2^{-8m}-close to uniform on (the entire set) B. With this in mind and noting that we have initialized $\varepsilon = 2^{-10m}$, we may now bound the probability of aborting:

At line 11 we have $\Pr[\mathbf{x} \in A \setminus A'] \leq \varepsilon + 2^{-8m} \ll 2^{-5m}$;

At line 21 we have $\Pr[\mathbf{x} \in A^{\dagger}] \leq \varepsilon + 2^{-8m} \ll 2^{-5m}$;

At line 25 we have $\Pr[\mathbf{y} \in \bigcup_{q \in B^{(0)}} B_q^{\dagger}] \leq \varepsilon + 2^{-8m} \ll 2^{-5m}$.

At line 32 we must notice that B decreases in size at three places in the algorithm: lines 17, 28 and 30. By property (c3) of the entropy-restoring partition (Lemma 4.8), in line 28 B will be cutoff by a total fraction which is no smaller than $(1-\varepsilon)2^{-|Q|} \geq 2^{-2Q} \geq 2^{-2000C} \geq 2^{-\frac{p}{20}+1}$ (more precisely, the prunning lemma says that this holds for each B_j, and hence it holds for the entire B). On the other hand, the probability that $|B|$ is reduced by a factor γ in line 17 is no greater than $\gamma 2^{6C} + C2^{-8m}$ — since it would be no greater than $\gamma 2^{6C}$ if $\mathbf{y}$ would be uniform on B (by Lemma 3.13), $\mathbf{y}$ is 2^{-8m}-close to uniform, and line 17 is executed C times (once for each bit Bob communicates in π). Hence $\gamma < 2^{-\frac{p}{20}}$ with probability at most $2^{-\frac{p}{20}+6C} + C2^{-8m} \ll 2^{-\frac{p}{100}}$; otherwise $|B|$ will be cuttoff by a fraction no smaller than $2^{-\frac{p}{10}+1}$, throughout the construction, in lines 17 and 28; In line 30, no more than $2^{-\frac{n}{10}} \ll 2^{-\frac{p}{10}+1}$ fraction is lost, and hence this implies that, with probability $\ll 2^{-\frac{p}{100}}$-close to 1, there must be at least $2^{\frac{9}{10}p}$-many q such that $|B_q| \geq 2^{\frac{9}{10}n}$ throughout the entire run of the protocol. This is the only point where the statistical distance can become greater than 2^{-5m}.

At line 35 we must notice the following: whenever Q gains k new elements — which only happens in line 29 — property (c4) of Lemma 4.8 says that the *affine-density* $\frac{|A|}{|\mathsf{Affine}^n(Q,u)|}$ will increase by $2^{\frac{1}{20}kn}$. Now, this affine density can only decrease in line 14, and as above the probability that this affine density is reduced by a factor γ is no greater than $\gamma \cdot 2^{6Cn} + Cn \cdot 2^{-8m}$. (The reason is similar to that in the previous paragraph — Alice's communication is Cn, and hence the first term comes form Lemma 3.13 assuming $\mathbf{x}$ is uniform on A, and the second term is from the fact that $\mathbf{x}$ is 2^{-8m} close to uniform.) For $\gamma \leq 2^{-10Cn}$, the probability that the affine-density of A decreases by a factor smaller than 2^{-10Cn} is $\leq \gamma \cdot 2^{6Cn} + Cn \cdot 2^{-8m} \ll 2^{-5m}$. And so, since the affine density is never greater than 1, it follows that $|Q|$ will remain bounded by $200 \cdot C$ with that much probability.

It then follows that $\bar{\pi}^{-1}(z)$ aborts with probability $\ll 2^{-5m}$, and hence the statistical distance between $\tilde{\pi}^{-1}(z)$ and $\bar{\pi}^{-1}(z)$ is $\ll 2^{-5m}$. It is easy to see that the statistical distance between $\tilde{\pi}^{-1}(z)$ and $\tilde{\tau}(z)$ is also small. The two processes behave the same, except in the way that A and B are updated; for example in line 13 of Algorithm 6, in order to choose $i \in \{0,1\}$, $\tilde{\pi}^{-1}(z)$ will choose uniformly-random tripple $(\mathbf{x}; \boldsymbol{q}, \mathbf{y}) \in A \times B$ such that $\mathbf{x} \cdot \mathbf{y} = z$, and then set $i = 0$ if $\mathbf{x} \in A' \cap A_{v_0}$, and set $i = 1$ if $\mathbf{x} \in A' \cap A_{v_1}$; $\tilde{\tau}(z)$ will do the same in line 11 of Algorithm 5, except $\mathbf{x}$ is uniformly chosen from A' instead. But by the inverse-marginals lemma (Lemma 4.9), the two resulting distributions on i are 2^{-8m}-close; more precisely, the lemma says that $\mathbf{x}$ as chosen π^{-1} is close to uniform conditioned on any fixed value for $\mathbf{q}$; but then it is also close to uniform overall. The same happens at every other point when the transcript is updated in both Algorithms 6 and 7. Since the length of the transcript is $O(Cn) \ll 2^p$, it follows that the total statistical distance between $\tilde{\pi}^{-1}(z)$ and $\tilde{\tau}(z)$ is $\ll 2^{-5m}$. The proof that $\tilde{\tau}(z)$ and $\tau(z)$ are close is almost identical to the proof that $\tilde{\tau}^{-1}(z)$ and $\tau^{-1}(z)$ are close.

This concludes the proof of Claim 6.8. We are left only to observe that the set Q contains the parity queries made to z, and $|Q|$ is forcefully bounded in lines 29–30 of Algorithm 7. So when we bounded the probability of the corresponding abort condition we also bounded the number of parity queries made by Algorithm 7.

Algorithm 6 Refined **deterministic** communication protocol $\bar{\pi}$ for VMV

input $x \in \{0,1\}^{p\times n}$ (to Alice) and $(q,y) \in \{0,1\}^p \times \{0,1\}^n$ (to Bob).
Both players initialize $A = \{0,1\}^{p\times n}$, $B = \{0,1\}^p \times \{0,1\}^n$.
Both players initialize $\varepsilon = 2^{-5p}$, $Q = \varnothing$, $w = \varnothing$, $u = \varnothing$.
Let v be the root of π.
while v is not a leaf **do**

Invariants which are true here are discussed in page 38.

  if $\mathsf{LH}^{\varepsilon}_{\infty}(A,Q) \geq \frac{9}{10}$ **then**
    Let v_0 and v_1 be the children of v.
    if Alice communicates at v **then**
      Let $A' \subseteq A$ have $\mathsf{LH}_{\infty}(A',Q) \geq \frac{9}{10}$ and $|A'| \geq (1-\varepsilon)|A|$.
      if $x \in A \setminus A'$ **then**
        Alice sends $\perp$. ▷ Extra bits
        Both players **run** $\pi_v(x,y)$ and **exit**.
      Alice sends i such that $x \in A_{v_i}$.
      Both players update $A = A' \cap A_{v_i}$ and $v = v_i$.
    else if Bob communicates at v **then**
      Bob sends i such that $y \in B_{v_i}$.
      Both players update $B = B_{v_i}$ and $v = v_i$.
  else if $\mathsf{LH}^{\varepsilon}_{\infty}(A,Q) < \frac{9}{10}$ **then**
    Apply Lemma 4.8 with $N = |B^{(0)}|$ and the various B_q, to obtain
    - A partition $A = A^{\dagger} \cup A_1 \cup A_2 \cdots$;
    - Partitions $B_q = B_q^{\dagger} \cup B'_q$ for each $q \in B^{(0)}$;
    - For each $i \geq 1$, the values of k'_i, Q'_i, and u'_i.
    if $x \in A^{\dagger}$ **then**
      Alice sends $\perp$. ▷ Extra bits
      Both players **run** $\pi_v(x,y)$ and **exit**.
    Alice sends i to Bob such that $x \in A_i$. ▷ Extra bits
    if $y \in B_q^{\dagger}$ for any $q \in B^{(0)}$ **then**
      Bob sends $\perp$. ▷ Extra bits
      Both players **run** $\pi_v(x,y)$ and **exit**.
    Bob sends $w'_i = u'_i \cdot y$ to Alice. ▷ Extra bits
    Both players update $A = A_i$, $B = B'(u'_i, w'_i)$, where $B' = \bigcup_{q\in B^{(0)}} B'_q$;
    Both players update $Q = Q \cup Q'_i$, $u = uu'_i$ and $w = ww'_i$.
  Remove from B every pair (q,y) such that $|B_q| < 2^{\frac{9}{10}n}$
  if $|B^{(0)}| \leq 2^{\frac{9}{10}p}$ **then**
    Bob sends $\perp$. ▷ Extra bits
    Both players **run** $\pi_v(x,y)$ and **exit**.
  if $|Q| > 200C$ **then**
    Alice sends $\perp$. ▷ Extra bits
    Both players **run** $\pi_v(x,y)$ and **exit**.
Output the label of the leaf v.

Algorithm 7 Randomized decision-tree procedure $\tau(z)$ for the impossible task

input $z \in \{0,1\}^p$.
Initialize $A = \{0,1\}^{p \times n}$, $B = \{0,1\}^p \times \{0,1\}^n$.
Initialize $\varepsilon = 2^{-5p}$, $Q = \varnothing$, $w = \varnothing$, $u = \varnothing$.
Let v be the root of π and $t = \varnothing$.
while v is not a leaf **do**

Invariants (i-iv) of page 38 hold here.

  if $\mathsf{LH}^{\varepsilon}_{\infty}(A, Q) \geq \frac{9}{10}$ **then**
    Let v_0 and v_1 be the children of v.
    if Alice communicates at v **then**
      Let $A' \subseteq A$ have $\mathsf{LH}_{\infty}(A', Q) \geq \frac{9}{10}$ and $|A'| \geq (1-\varepsilon)|A|$.
      With probability $\frac{|A \setminus A'|}{|A|}$, **abort**.
      Choose $i \in \{0,1\}$ with probability $\frac{|A' \cap A_{v_i}|}{|A'|}$ and set $t = ti$.
      Update $A = A'_{v_i}$ and $v = v_i$.
    else if Bob communicates at v **then**
      Choose $i \in \{0,1\}$ with probability $\frac{|B \cap B_{v_i}|}{|B|}$ and set $t = ti$.
      Update $B = B_{v_i}$ and $v = v_i$.
  else if $\mathsf{LH}^{\varepsilon}_{\infty}(A, Q) < \frac{9}{10}$ **then**
    Apply Lemma 4.8 with $N = |B^{(0)}|$ and the various B_q, to obtain
    - A partition $A = A^{\dagger} \cup A_1 \cup A_2 \cdots$;
    - Partitions $B_q = B_q^{\dagger} \cup B'_q$ for each $q \in B^{(0)}$;
    - For each $i \geq 1$, the values of k'_i, Q'_i, and u'_i.

    With probability $\frac{|A^{\dagger}|}{|A|}$, **abort**.
    Choose $i \geq 1$ with probability $\frac{|A_i|}{|A \setminus A^{\dagger}|}$
    Set $t = t\ i$.
    With probability $\frac{|\cup_q B_q^{\dagger}|}{|B|}$, **abort**.
    Query z to discover $w'_i = u'_i \cdot z$.
    Set $t = t\ w'_i$.
    Update $A = A_i$, $B = B'(u'_i, w'_i)$, where $B' = \bigcup_{q \in B^{(0)}} B'_q$;
    Update $Q = Q \cup Q'_i$, $u = uu'_i$ and $w = ww'_i$.
  Remove from B every pair (q, y) such that $|B_q| < 2^{\frac{9}{10}n}$
  if $|B^{(0)}| \leq 2^{\frac{9}{10}p}$ **then**
    Abort.
  if $|Q| > 200C$ **then**
    Abort.
Choose a uniformly-random $(q, y) \in B$;
if q is *not* linearly-independent of Q **then**
  Abort
else
  Output (q, t).

□

References

[ADKF70] V. Z. Arlazarov, E. A. Dinic, M. A. Kronrod, and I. A. Faradzev. On economical construction of the transitive closure of a directed graph. *Soviet Mathematics Doklady*, pages 11(5):1209–1210, 1970.

[AGJ+17] Anurag Anshu, Naresh B Goud, Rahul Jain, Srijita Kundu, and Priyanka Mukhopadhyay. Lifting randomized query complexity to randomized communication complexity. *arXiv:1703.07521*, 2017.

[AGM03] Noga Alon, Oded Goldreich, and Yishay Mansour. Almost k-wise independence versus k-wise independence. *Information Processing Letters*, 88(3):107–110, 2003.

[Ajt88] Miklós Ajtai. A lower bound for finding predecessors in yao's cell probe model. *Combinatorica*, 8(3):235–247, 1988.

[BHP10] Paul Beame, Trinh Huynh, and Toniann Pitassi. Hardness amplification in proof complexity. In *Proceedings of the 42nd STOC*, pages 87–96, 2010.

[BIPW10] Khanh Do Ba, Piotr Indyk, Eric Price, and David P. Woodruff. Lower bounds for sparse recovery. In *Proceedings of the 21st SODA*, pages 1190–1197, 2010.

[Bra11] Mark Braverman. Poly-logarithmic independence fools bounded-depth boolean circuits. *Commun. ACM*, 54(4):108–115, 2011.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.

[CGL15] Raphael Clifford, Allan Grønlund, and Kasper Green Larsen. New unconditional hardness results for dynamic and online problems. In *Proceedings of the 56th FOCS*, pages 1089–1107, 2015.

[Cha07] Arkadev Chattopadhyay. Discrepancy and the power of bottom fan-in in depth-three circuits. In *Proceedings of the 48th FOCS*, pages 449–458, 2007.

[Cha08] Arkadev Chattopadhyay. *Ciruits, Communication and Polynomials*. PhD thesis, McGill University, 2008.

[CKLM17] Arkadev Chattopadhyay, Michal Kouckỳ, Bruno Loff, and Sagnik Mukhopadhyay. Simulation theorems via pseudorandom properties. *arXiv:1704.06807*, 2017.

[CR12] Amit Chakrabarti and Oded Regev. An optimal lower bound on the communication complexity of gap-hamming-distance. *SIAM Journal on Computing*, 41(5):1299–1317, 2012.

[CW16] Timothy M. Chan and Ryan Williams. Deterministic APSP, orthogonal vectors, and more: Quickly derandomizing razborov-smolensky. In *Proceedings of the 27th SODA*, pages 1246–1255, 2016.

[DL17] Holger Dell and John Lapinskas. Fine-grained reductions from approximate counting to decision. *CoRR*, abs/1707.04609, 2017.

[DPV09] Matei David, Toniann Pitassi, and Emanuele Viola. Improved separations between nondeterministic and randomized multiparty communication. *ACM Transactions on Computation Theory*, 1(2), 2009.

[FHM01] Gudmund Skovbjerg Frandsen, Johan P Hansen, and Peter Bro Miltersen. Lower bounds for dynamic algebraic problems. *Information and Computation*, 171(2):333–349, 2001.

[GKPW17] Mika Göös, Pritish Kamath, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for P^{np}. In *Proceedings of the 32nd CCC*, 2017.

[GLM+15] Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. In *Proceedings of the 47th STOC*, pages 257–266, 2015.

[GP14] Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. In *Proceedings of the 46th STOC*, pages 847–856, 2014.

[GPW15] Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. In *Proceedings of the 56th FOCS*, 2015.

[GPW17] Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for bpp. In *Proceedings of the 58th FOCS*, 2017.

[HHL16] Hamed Hatami, Kaave Hosseini, and Shachar Lovett. Structure of protocols for xor functions. In *Proceedings of the 57th FOCS*, pages 282–288, 2016.

[HKNS15] Monika Henzinger, Sebastian Krinninger, Danupon Nanongkai, and Thatchaphol Saranurak. Unifying and strengthening hardness for dynamic problems via the online matrix-vector multiplication conjecture. In *Proceedings of the 47th STOC*, pages 21–30, 2015.

[HN12] Trinh Huynh and Jakob Nordstrom. On the virtue of succinct proofs: Amplifying communication complexity hardness to time-space trade-offs in proof complexity. In *Proceedings of the 44th STOC*, pages 233–248, 2012.

[JKKR04] T. S. Jayram, Subhash Khot, Ravi Kumar, and Yuval Rabani. Cell-probe lower bounds for the partial match problem. *J. Comput. Syst. Sci.*, 69(3):435–447, 2004.

[Joh01] Jan Johannsen. Depth lower bounds for monotone semi-unbounded fan-in circuits. *ITA*, 35(3):277–286, 2001.

[Jus72] Jørn Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18(5):652–656, 1972.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, New York, NY, USA, 1997. 0.

[KW90] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM J. Discrete Math.*, 3(2):255–265, 1990.

[Lar12] Kasper Green Larsen. Higher cell probe lower bounds for evaluating polynomials. In *Proceedings of the 53rd FOCS*, pages 293–301, 2012.

[LW17] Kasper Green Larsen and Ryan Williams. Faster online matrix-vector multiplication. In *Proceedings of the 28th SODA*, pages 2182–2189, 2017.

[LWY17] Kasper Green Larsen, Omri Weinstein, and Huacheng Yu. Crossing the logarithmic barrier for dynamic boolean data structure lower bounds. *arXiv:1703.03575*, 2017.

[Mil94] Peter Bro Miltersen. Lower bounds for union-split-find related problems on random access machines. In *Proceedings of the 26th STOC*, pages 625–634, 1994.

[MNSW98] 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.

[MP17] Raghu Meka and Toniann Pitassi, editors. *Hardness Escalation in Communication Complexity and Query Complexity, Workshop at 58th FOCS*, 2017.

[Pat11] Mihai Patrascu. Unifying the landscape of cell-probe lower bounds. *SIAM Journal on Computing*, 40(3):827–847, 2011.

[PT06] Mihai Patrascu and Mikkel Thorup. Time-space trade-offs for predecessor search. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 232–240, 2006.

[PT09] Mihai Patrascu and Mikkel Thorup. Higher lower bounds for near-neighbor and further rich problems. *SIAM J. Comput.*, 39(2):730–741, 2009.

[RM99] Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, 1999.

[RPRC16] Robert Robere, Toniann Pitassi, Benjamin Rossman, and Stephen A Cook. Exponential lower bounds for monotone span programs. In *Proceedings of the 57th FOCS*, pages 406–415, 2016.

[RR15] Sivaramakrishnan Natarajan Ramamoorthy and Anup Rao. How to compress asymmetric communication. In *Proceedings of the 30th CCC*, pages 102–123, 2015.

[RW04] Renato Renner and Stefan Wolf. Smooth rényi entropy and applications. In *International Symposium on Information Theory*, page 233. IEEE, 2004.

[She09] Alexander A. Sherstov. Separating AC0 from depth-2 majority circuits. *SIAM Journal on Computing*, 38(6):2113–2129, 2009.

[She12] Alexander A. Sherstov. The communication complexity of gap hamming distance. *Theory of Computing*, 8(1):197–208, 2012.

[Sok17] Dmitry Sokolov. Dag-like communication and its applications. In *International Computer Science Symposium in Russia*, pages 294–307, 2017.

[SV08] 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.

[Vaz86] Umesh Vazirani. *Randomness, Adverseries and Computation.* PhD thesis, University of California, Berkeley, 1986.

[Vid12] Thomas Vidick. A concentration inequality for the overlap of a vector on a large set. *Chicago Journal OF Theoretical Computer Science*, 1:1–12, 2012.

[Wat17] Thomas Watson. A $\mathsf{ZPP}^{\mathsf{NP}}$ lifting theorem. *Unpublished preprint*, 2017.

[Wil07] Ryan Williams. Matrix-vector multiplication in sub-quadratic time (some preprocessing required). In *Proceedings of the SODA*, volume 7, pages 995–1001, 2007.

[Woo14] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1-2):1–157, 2014.

[WY14] Ryan Williams and Huacheng Yu. Finding orthogonal vectors in discrete structures. In *Proceedings of the 25th SODA*, pages 1867–1877, 2014.

[WYY17] Xiaodi Wu, Penghui Yao, and Henry S. Yuen. Raz-mckenzie simulation with the inner product gadget. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:10, 2017.

[Xia92] B. Xiao. *New bounds in cell probe model*. PhD thesis, UC San Diego, 1992.

[Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the 11h STOC*, pages 209–213, 1979.

[Yao15] Penghui Yao. Parity decision tree complexity and 4-party communication complexity of XOR-functions are polynomially equivalent. *arXiv:1506.02936*, 2015.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il