Electronic Colloquium on Computational Complexity, Report No. 84 (2023)

Time-Space Lower Bounds for Bounded-Error Computation in the Random-Query Model

Itai Dinur

May 31, 2023

Abstract

The random-query model was introduced by Raz and Zhan at ITCS 2020 as a new model of space-bounded computation. In this model, a branching program of length T and width 2^S attempts to compute a function $f : \{0,1\}^n \to \{0,1\}$. However, instead of receiving direct access to the input bits $(x_1, \ldots, x_n)$, the input is given in pairs of the form $(i_j, x_{i_j}) \in \{1, \ldots, n\} \times \{0,1\}$ for $j = 1, 2, \ldots, T$, where the indices $i_1, \ldots, i_T$ are chosen at random from a pre-fixed distribution.

Raz and Zhan proved that any branching program in the random-query model with the independent distribution (where $\{i_j\}_{j=1,\ldots,T}$ are uniform and independent) that computes a function f with sensitivity k satisfies $T \cdot (S + \log n) \geq \Omega(n \cdot k)$. This gives a quadratic time-space lower bound for many natural functions which have sensitivity $\Omega(n)$, such as XOR and Majority. The bound was proved in the zero-error regime, where for each input, the branching program is required to output a value with high probability, and given that a value is output, it must be correct with probability 1.

Furthermore, Raz and Zhan conjectured that (up to logarithmic factors in n) a quadratic time-space lower bound still holds for the XOR function in the more conventional bounded-error regime, where for each input, the output must be correct with high probability.

In this paper, we prove this conjecture. More generally, let $f : \{0,1\}^n \to \{0,1\}$ have *average sensitivity* (or total influence) $I[f]$. We prove that any branching program in the random-query model with the independent distribution that computes f in the bounded-error regime satisfies $T \cdot S \geq \tilde{\Omega}(n) \cdot I[f]$ (where $\tilde{\Omega}$ hides logarithmic factors in n). Moreover, we prove a quadratic time-space lower bound for the Majority function, even though its total influence is $\Theta(\sqrt{n})$.

Our proof is based on a reduction from a communication complexity problem.

1 Introduction

The study of time-space tradeoffs aims at understanding what can be computed efficiently with limited space. At ITCS 2020 [RZ20], Raz and Zhan introduced the random-query model for studying time-space tradeoff lower bounds. In the main use-case, the goal of a branching program (which models a non-uniform algorithm[1]) is to compute a function $f : \{0,1\}^n \to \{0,1\}$ with minimal time T (i.e., length, measured in queries to input bits) using space limited to S bits (i.e., width 2^S) at any stage of the computation. However, the branching program does not have direct access to

Department of Computer Science, Ben-Gurion University, Israel. dinuri@bgu.ac.il.

[1] See Section 2 for a formal definition of branching programs.

ISSN 1433-8092

the input $x = (x_1, \ldots, x_n) \in \{0,1\}^n$, but rather at each stage of the computation $j = 1, \ldots, \mathrm{T}$, it receives a pair $(i_j, x_{i_j}) \in [n] \times \{0,1\}$ (where $[n] = \{1, \ldots, n\}$), such that i_j is a query index chosen at random according to a pre-defined distribution.

Raz and Zhan described several motivations for the random-query model. First, it is a natural and interesting model of computation in its own right. Moreover, the model is related to the recent line of works on proving time-space lower bounds for learning [SVW16, MM18, BGY18, GRT19, Raz19]. Specifically, the goal of a branching program in this model is to distinguish between the input sets $\{x \mid f(x) = 0\}$ and $\{x \mid f(x) = 1\}$, where the query indices $i_1, \ldots, i_{\mathrm{T}}$ are selected from the *independent distribution*, namely, they are mutually independent uniform random variables, viewed as samples. Another distribution considered in [RZ20] is the *recurring distribution*, in which the only dependencies among $i_1, \ldots, i_{\mathrm{T}}$ are equalities. As shown by Raz and Zhan, under the recurring distribution, the random-query model is closely related to the standard model of oblivious branching programs. Proving slightly super-linear time-space tradeoff lower bounds for oblivious branching programs is a long standing open problem (see [BNS89] for the best-known result), providing additional motivation for studying the random-query model.

The main technical result of [RZ20] is a proof that any branching program in the random-query model with the independent distribution that computes a function $f : \{0,1\}^n \to \{0,1\}$ with (maximal) sensitivity[2] k using space S and time T satisfies $\mathrm{T} \cdot (\mathrm{S} + \log n) \geq \Omega(n \cdot k)$. In particular, this gives a quadratic time-space lower bound for many natural functions which have maximal sensitivity $\Omega(n)$ (such as XOR and Majority). Yet, the result was proved in the zero-error regime, where for every input, the branching program is allowed not to output a (Boolean) value with probability at most $1/2$, but given that a value is output, it must be correct with probability 1. On the other hand, proving time-space lower bounds in the more standard bounded-error regime (where for every input, the output must be correct with high probability) was left open.

As noted in [RZ20], the zero-error regime is in general substantially different from the bounded-error regime in the random-query model. For example, it is possible to compute the AND function using $\mathrm{S} \leq O(1)$ and $\mathrm{T} \leq O(n)$ simply be outputting 1 if and only if $x_{i_j} = 1$ for all $j = 1, \ldots, T$. Yet, the maximal sensitivity of AND is n, and thus in the zero-error regime, $\mathrm{T} \cdot (\mathrm{S} + \log n) \geq \Omega(n^2)$. In contrast, computing the XOR function with a small amount of space still seems to be hard in the bounded-error regime. Consequently, Raz and Zhan made the following conjecture.

Conjecture 1 ([RZ20])**.** Under the random-query model with the independent distribution, any branching program of length T (time T) and width 2^{S} (space S) which computes $x_1 \oplus \ldots \oplus x_n$ with error $1/3$ must satisfy $\mathrm{T} \cdot \mathrm{S} = \tilde{\Omega}(n^2)$.

Here, $\tilde{\Omega}$ hides logarithmic factors in n.

Remark 1. There is a branching program that computes the XOR function with $\mathrm{T} \cdot \mathrm{S} = \tilde{O}(n^2)$. For example, if $\mathrm{S} = \tilde{O}(1)$, then it computes the XOR by considering the input bits in order $x_1, x_2, \ldots, x_n$, waiting an average of $O(n)$ time until each (i, x_i) arrives. Therefore, proving Conjecture 1 would establish an essentially tight time-space tradeoff for XOR in the bounded-error regime.

Remark 2. Proving a (slightly) super-quadratic time-space tradeoff lower bound for some function $f : \{0,1\}^n \to \{0,1\}$ in the random-query model would resolve the long standing open problem

[2]The sensitivity of f on x is the number of pivotal coordinates of x, namely $\mathrm{sens}_f(x) = |\{j \mid f(x_1, \ldots, x_n) \neq (x_1, \ldots, x_{j-1}, x_j \oplus 1, x_{j+1}, \ldots, x_n)\}|$. The maximal sensitivity is $\max_x\{\mathrm{sens}_f(x)\}$.

of proving a (slightly) super-linear time-space tradeoff lower bound for arbitrary (non-oblivious) branching programs (see [BSSV03] for the best-known result). This follows since any query of a branching program to an input bit can be simulated using $O(n)$ queries on average in the random-query model. In other words, barring a significant breakthrough, a quadratic time-space lower bound (as in Conjecture 1) is essentially the best we can hope to prove for any function.

1.1 Our Results

In this paper, we consider a function $f : \{0,1\}^n \to \{0,1\}$ with total influence $\mathrm{I}[f]$ (which is equal to its *average sensitivity*). We prove the following theorem (see Theorem 2 for the formal version).

Theorem 1 (Informal)**.** Under the random-query model with the independent distribution, any branching program of length T (time T) and width 2^{S} (space S) which computes $f : \{0,1\}^n \to \{0,1\}$ with constant error $\delta < 1/2$ must satisfy

$$\mathrm{T} \cdot \mathrm{S} \geq \tilde{\Omega}(n) \cdot \mathrm{I}[f].$$

Moreover, if $\mathrm{I}[f] \geq \Omega(n)$ then

$$\mathrm{T} \cdot \mathrm{S} \geq \Omega(n^2).$$

Since the XOR function has total influence n, then Conjecture 1 is a special case of the theorem.

Theorem 1 can be considered as an analog of the main result of [RZ20] in the zero-error regime, which is parameterized on the maximal sensitivity of f. The theorem is tight since the partial XOR function $f(x_1, \ldots, x_n) = x_1 \oplus \ldots \oplus x_k$ for $k \in [n]$ has total influence k, while there is a branching program that computes it with $\mathrm{T} \cdot \mathrm{S} \leq \tilde{O}(n) \cdot k$. We further note that any non-constant function f requires $\mathrm{T} \geq \Omega(n)$ to compute with bounded-error. Hence, when $\mathrm{I}[f] \leq o(1)$, the theorem is not tight, but the trivial tradeoff formula $\mathrm{T} \cdot \mathrm{S} \geq \tilde{\Omega}(n)$ which is independent of the influence is trivially tight for some function with total influence $\mathrm{I}[f] \leq o(1)$ (e.g. an AND of an appropriately chosen number of bits).

While Theorem 1 gives a tight time-space lower bound for many natural functions, it does not seem to be tight for others. A notable example is the Majority function, which has total influence of $\Theta(\sqrt{n})$ [O'D14]. Yet, for this function, we improve the general result of Theorem 1 and show an essentially tight bound of $\mathrm{T} \cdot \mathrm{S} \geq \Omega(n^2)$.

1.2 Our Techniques

Raz and Zhan obtained their results by directly analyzing the properties of branching programs in the zero-error regime. Specifically, they devised a reduction to computing a function f from a variant of the coupon collector problem, where the goal of the branching problem is to declare when it has seen all indices of $[n]$ in $i_1, i_2, \ldots$.

We take a different approach and prove Theorem 1 by a reduction from communication complexity. Below, We give an informal overview of our proof.

1.2.1 The Missing-Element communication problem.

Consider the following number-in-hand multiparty communication problem, which we call the *Missing-Element problem*. For $k \gg 2$, there are k players, where player $i \in [k]$ receives input

vector $X^i \in \{0,1\}^n$, viewed as an indicator of a subset of $[n]$. The vectors $X = X^1, \ldots, X^k$ are distributed as follows. With probability $1/2$, each X^i is independently uniformly distributed. On the other hand, with probability $1/2$, there is some common zero entry (element) $j \in [n]$ in all vectors, namely, for all $i \in [k]$, $X^i_j = 0$, while the remaining $n-1$ bits are independently uniformly distributed. We think of $j \in [n]$ as uniformly chosen. The goal of the players is to distinguish between these two distributions with high probability by writing (communicating) a minimal number of bits on a shared blackboard.

Remark 3. The definition of the Missing-Element problem is non-standard, as the goal of the players is to distinguish between two specific distributions with intersecting supports, rather than to compute a function of their inputs. This definition is tailored towards our reduction to computing a function in the random-query model with the independent distribution.

Remark 4. One can view the Missing-Element problem as a "communication analog" of the coupon collector problem, since the players are trying to decide whether they can jointly "collect all the coupons" in $[n]$.

We prove that regardless of the value of k, any protocol that solves the Missing-Element problem with high probability must communicate $\Omega(n)$ bits for some input. The proof uses a variant of the information complexity technique [CSWY01, BJKS04], which is a standard method for proving communication complexity lower bounds, particularly for the classical (multiparty) Set-Disjointness problem. In fact, the Missing-Element problem is closely related to the multiparty Set-Disjointness problem, since if all players flip the bits of their inputs, their goal would be to detect whether they have a common element, as in Set-Disjointness. Yet, the Missing-Element problem is defined for a specific distribution. As far as we know the result that we prove for the specific distribution of the Missing-Element problem has not been established before.

We stress that our goal is to analyze the *distributional* communication complexity of the Missing-Element problem, namely, the error probability is taken over the random coins of the players, as well as the random choice of input under the distribution. This stands in contrast to several related works such as [BJKS04, Gro09, Jay09, BEO+13] that analyze the *randomized* communication complexity of (multiparty) Set-Disjointness, where the error probability is taken over the random coins of the players, but is worst-case over the input. Specifically, these works aim to prove that any protocol whose communication cost is too small fails with high probability on *some particular input* (even if the probability of obtaining this input under the analyzed distribution is minuscule). Such analysis is insufficient in our distributional setting.

1.2.2 Proving time-space lower bounds in the random-query model.

Next, we reduce the problem of computing $f : \{0,1\}^n \to \{0,1\}$ in the random-query model with the independent distribution in the bounded-error regime from the Missing-Element problem. For simplicity, assume first that f is the XOR function.

Suppose that there is a branching program $\mathcal{P}$ that computes the XOR function with high probability on every n-bit input in the random-query model. The k parties use $\mathcal{P}$ to solve the Missing-Element problem as follows. Each player $i \in [k]$ uses its n-bit input X^i to generate a subset of $\Theta(n)$ uniformly distributed query indices in $[n]$ (assuming there is no missing index), which will be fed to $\mathcal{P}$. It remains to define the actual input to $\mathcal{P}$, which must be consistent across all parties. For this

purpose, we crucially observe that the $\Omega(n)$ communication lower bound for the Missing-Element problem remains almost the same even for *public-coin protocols*. Namely, we may assume that the players share a common random string chosen independently of the input, but can essentially be ignored when calculating the total communication cost. In our reduction, the players share an n-bit string chosen uniformly at random and denoted by $(R_1, \ldots, R_n) \in \{0,1\}^n$. This string serves as the input to $\mathcal{P}$, which should return $\oplus_{i=1}^n R_i$.

The reduction protocol works as follows. The first player uses its input to generate $\Theta(n)$ query indices and the corresponding "answers" using $(R_1, \ldots, R_n)$ and feeds them to $\mathcal{P}$. Player 1 then writes the resultant intermediate state on the blackboard using S bits. Next, the second player continues the execution of $\mathcal{P}$ using its generated query indices and answers, and so forth. Finally, the last player finishes the execution of $\mathcal{P}$ and computes its output. In the uniform distribution (with no missing element), $\mathcal{P}$ is executed in the random-query model with the independent distribution and thus returns $\oplus_{i=1}^n R_i$ with high probability. On the other hand, in the distribution where there is a missing element $j \in [n]$, we have no formal guarantee about the behaviour of $\mathcal{P}$. Yet, $\mathcal{P}$ is clearly independent of the uniform bit R_j, and thus outputs $\oplus_{i=1}^n R_i$ with probability $1/2$. The constant gap between the success probabilities of $\mathcal{P}$ (depending on the input distribution of the players) allows the players to solve the Missing-Element problem with high probability.

The total communication of the protocol is $\Theta(k \cdot \mathrm{S})$, and thus by our proof regarding the hardness of the Missing-Element problem $k \cdot \mathrm{S} \geq \Omega(n)$, or $\mathrm{S} \geq \Omega(n)/k$. Moreover, the number of query indices generated by the players and fed to $\mathcal{P}$ is $\mathrm{T} \geq \Omega(n) \cdot k$. Therefore, $\mathrm{T} \cdot \mathrm{S} \geq \Omega(n^2)$ for the XOR function, as claimed in Theorem 1.

For arbitrary f, the gap between the success probabilities of $\mathcal{P}$ (depending on the input distribution of the players) is a function of $\mathrm{I}[f]$. Our general reduction protocol obtains constant success probability for the Missing-Element problem using a careful composition of amplification procedures, resulting in the essentially tight time-space lower bound of Theorem 1. This bound avoids the standard quadratic loss associated with amplification using the Chernoff bound.[3]

For the Majority function, we define a non-uniform distribution on $(R_1, \ldots, R_n)$ which increases the gap in the success probabilities of $\mathcal{P}$ compared to the uniform distribution, and allows to obtain an essentially tight time-space lower bound.

Remark 5. It follows from the proof that our time-space lower bounds hold even if the space (width) of the branching program is bounded only in time intervals of (sufficiently small) length $\Theta(n)$.

1.3 Paper Structure

We give preliminaries in Section 2. In Section 3 we define and analyze the Missing-Element multiparty communication problem, while in Section 4 we use this analysis to prove time-space lower bounds.

[3] The formal version of Theorem 1 (Theorem 2) depends quasi-linearly on $\mathrm{I}[f]$.

2 Preliminaries

Unless stated otherwise, the parameters of the protocols and branching programs we consider are functions of the instance size, parameterized by n. We use capital letters to denote random variables and lower case letters to denote values they attain.

For a positive integer n, let $[n] = \{1, 2, \ldots, n\}$.

We use the following (special case of) Hoeffding's inequality.

Proposition 1 (Hoeffding's inequality)**.** Let $Z_1, \ldots, Z_t \in \{0, 1\}$ be independent random variables and $M = \sum_{i=1}^{t} Z_i$. Then, for any $\alpha \geq 0$,

$$\Pr[M - \mathrm{E}[M] > \alpha] \leq \exp(-2\alpha^2/t).$$

2.1 Boolean Functions

We review several basic notions in analysis of Boolean function. Refer to [O'D14] for more details.

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. We say that a coordinate $j \in [n]$ is pivotal for f on input $x = (x_1, \ldots, x_n)$ if $f(x) \neq f(x^{\oplus j})$, where $x^{\oplus j} = (x_1, \ldots, x_{j-1}, x_j \oplus 1, x_{j+1}, \ldots, x_n)$. The sensitivity of f at x (denoted $\mathrm{sens}_f(x)$) is the number of pivotal coordinates of f on input x.

The influence of coordinate j on f is $\mathrm{Inf}_j[f] = \Pr_X[f(X) \neq f(X^{\oplus j})]$. The total influence of f is $\mathrm{I}[f] = \sum_{j=1}^{n} \mathrm{Inf}_j[f]$.

Proposition 2.

$$\mathrm{I}[f] = n \cdot \Pr_{J,X}[J \text{ is pivotal for } f \text{ on } X].$$

Proof. We have

$$\begin{aligned}
\mathrm{I}[f] &= n \cdot \mathop{\mathrm{E}}_{J}[\mathrm{Inf}_J[f]] \\
&= n \cdot \mathop{\mathrm{E}}_{J}[\Pr_X[f(X) \neq f(X^{\oplus J})]] \\
&= n \cdot \mathop{\mathrm{E}}_{J,X}[\ f(X) \neq f(X^{\oplus J})\ ] \\
&= n \cdot \Pr_{J,X}[J \text{ is pivotal for } f \text{ on } X].
\end{aligned}$$

■

Furthermore, by a similar argument $\mathrm{I}[f] = \mathrm{E}_X[\mathrm{sens}_f(X)]$.

2.2 Models of Computation

We define the branching program model of computation and the random-query model with the independent distribution, as defined in [RZ20].

A branching program of length T and width 2^{S} is a directed graph with vertices arranged in $\mathrm{T}+1$ layers, each containing at most 2^{S} vertices. In layer 0 there is one vertex, called the start vertex. In the last layer (layer T) each vertex has out-degree 0, and is called a leaf. For every $i < \mathrm{T}$, the outgoing edges from every non-leaf vertex in layer i only go to vertices in layer $i+1$.

In a branching program for computing a function $f : \{0,1\}^n \to \{0,1\}$, every non-leaf vertex has $2n$ outgoing edges, labeled once with each element in $[n] \times \{0,1\}$. Every leaf v in the program is labeled with an output $\tilde{f}_v \in \{0,1\}$. Given an input $x \in \{0,1\}^n$ and indices $i_1, \ldots, i_{\mathrm{T}} \in [n]$, the computation path in the branching program starts from the start vertex, and at step j follows the edge labeled with (i_j, x_{i_j}) until reaching a leaf v, and outputs $\tilde{f}_v$.

In the *random-query model with the independent distribution*, the indices $i_1, \ldots, i_{\mathrm{T}}$ are chosen independently and uniformly at random.

We say that the branching program computes $f : \{0,1\}^n \to \{0,1\}$ with error δ, if for every $x \in \{0,1\}^n$, the probability that the output of the branching program satisfies $\tilde{f}_v = f(x)$ is at least $1 - \delta$. In the random-query model, the probability is taken over the choice of the random variables $I_1, \ldots, I_{\mathrm{T}}$.

2.3 Communication Complexity

We consider a number-in-hand multiparty communication model in which the input is partitioned amongst several players, and their goal is to compute some function by exchanging messages via a shared blackboard. For a protocol Π and input x we call the concatenation of all messages written on the blackboard the transcript of Π on x, and denote it by $\Pi(x)$. At the end of the protocol an output function Π_{out} is applied to $\Pi(x)$ and the execution of the protocol succeeds if it correctly computes the function. We remark that in this paper we allow the function to depend on variables that are not given as input to the players.

In a private-coin randomized protocol, each player has private access to random coins, while in a public-coin randomized protocol, the players also have access to a shared public random string, written on the blackboard. For randomized protocols, $\Pi(x)$ is a random variable, and we denote by Π_x its distribution. We further note that we allow Π_{out} to be randomized. The communication cost of a protocol Π is defined as the maximum length of $\Pi(x)$ over all valid inputs x and over the random coins.

For a public-coin protocol Π, we also define the quantity $\mathrm{C}_v(\Pi)$ as the maximum length of $\Pi(x)$ over all valid inputs x and over the random coins, excluding the shared public random string. For this purpose, we think of the shared string as written on a separate blackboard, which is not taken into account when calculating $\mathrm{C}_v(\Pi)$.

In this paper we will generally be interested in the error probability of protocols, where the probability is taken over both the input (whose distribution is defined according to the specific problem), as well as the independent randomness used by the players.

2.4 Information Theory

We review some definitions and facts from information theory. Refer to [CT06] for more details.

Let X, Y, Z be random variables. Given that X is defined on domain Ω, we denote by $H(X) = \sum_{\omega \in \Omega} \Pr[X = \omega] \log \frac{1}{\Pr[X=\omega]}$ its Shannon entropy, and by

$$H(X \mid Y) = \mathop{\mathrm{E}}_{y}[H(X \mid Y = y)]$$

the conditional entropy of X given Y.

The mutual information between X and Y is defined as

$$I(X;Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X).$$

The conditional mutual information between X and Y conditioned on Z is

$$I(X;Y \mid Z) = H(X \mid Z) - H(X \mid Y,Z) = \mathop{\mathrm{E}}_{z}[I(X;Y \mid Z = z)].$$

We state several basic properties that we use. First, conditioning cannot increase entropy, namely $H(X \mid Y) \leq H(X)$, with equality if and only if X and Y are independent. This implies that the (conditional) mutual information is always non-negative.

For random variables $X_1, \ldots, X_n, Y$, the chain rule for mutual information asserts that

$$I(X_1, \ldots, X_n; Y) = \sum_{i=1}^{n} I(X_i; Y \mid X_{<i}),$$

where $X_{<i} = X_1, \ldots, X_{i-1}$ (if $i = 1$ the sequence is empty).

2.5 Hellinger Distance

The Hellinger distance between distributions P and Q on domain Ω is defined as

$$h(P,Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{\omega \in \Omega} |\sqrt{P(\omega)} - \sqrt{Q(\omega)}|^2}.$$

We state several useful properties of the Hellinger distance. These are taken from [BJKS04], and we adapt them to our purpose.

Proposition 3 ([BJKS04], Lemma 6.5)**.** Let P_0, P_1 be two distributions. Assume that X_Z is generated by first picking $Z \in_R \{0,1\}$ uniformly at random and then sampling from P_Z. Then,

$$I(Z; X_Z) \geq h^2(P_0, P_1).$$

Proposition 4 ([BJKS04], Lemma 6.2)**.** Let $\delta_0, \delta_1 > 0$ be parameters. Let Π be a protocol and let x, y be inputs to Π such that there exists an output z for which $\Pr[\Pi(x) = z] \geq 1 - \delta_0$ and $\Pr[\Pi(y) = z] \leq \delta_1$. Then,

$$h^2(\Pi_x, \Pi_y) \geq 1 - \sqrt{2(\delta_0 + \delta_1)}.$$

Proposition 5 ([BJKS04], Lemma 6.3)**.** Let Π be a k-party private-coin protocol and let $x = (x^1, \ldots, x^k), y = (y^1, \ldots, y^k)$ be inputs to Π. Let $x' = ((x')^1, \ldots, (x')^k), y' = ((y')^1, \ldots, (y')^k)$ be obtained from x, y by performing "cut-and-paste", namely, for each $i \in [k]$, either $(x')^i = x^i$ and $(y')^i = y^i$, or $(x')^i = y^i$ and $(y')^i = x^i$. Then,

$$h(\Pi_x, \Pi_y) = h(\Pi_{x'}, \Pi_{y'}).$$

Proposition 6 ([BJKS04], Lemma 6.4)**.** Let Π be a k-party private-coin protocol and let $x = (x^1, \ldots, x^k), y = (y^1, \ldots, y^k)$ be inputs to Π. Let $s \subset [k]$. Let $x' = ((x')^1, \ldots, (x')^k)$ be obtained from x, y, s by setting $(x')^i = x^i$ for all $i \in s$ and $(x')^i = y^i$ for all $i \notin s$. Let $y' = ((y')^1, \ldots, (y')^k)$ be generated from x, y, s by setting $(y')^i = y^i$ for all $i \in s$ and $(y')^i = x^i$ for all $i \notin s$. Then,

$$h^2(\Pi_x, \Pi_{x'}) + h^2(\Pi_{y'}, \Pi_y) \leq 2h^2(\Pi_x, \Pi_y).$$

We remark that propositions 5 and 6 were proved in [BJKS04] only for $k = 2$, but their generalization to $k > 2$ is straightforward.

3 The Missing-Element Communication Problem

3.1 The Unit Missing-Element Communication Problem

The multiparty (unit) Missing-Element communication problem (ME_k) is a problem with k players. The input to the problem is denoted $X = (X^1, \ldots, X^k) \in \{0,1\}^k$, where each player $i \in [k]$ gets input bit $X^i \in \{0,1\}$. At the beginning of the game a bit $B \in \{0,1\}$ is chosen uniformly at random. B determines the input distribution as follows.

If $B = 0$, then $X^i = 0$ for all $i \in [k]$.

If $B = 1$, each X^i is selected independently and uniformly at random.

Let Π be a protocol for the ME_k problem. We define several types of error probabilities.

For $b \in \{0,1\}$, let $\delta_b(\Pi) = \Pr[\Pi_{\mathrm{out}}(\Pi(X)) \neq b \mid B = b]$.

The overall error probability of Π is $\delta(\Pi) = \mathrm{E}_B[\delta_B(\Pi)] = (\delta_0(\Pi) + \delta_1(\Pi))/2$.

We stress that the error probabilities are taken over all the random variables defined, including the input X. Note that for any protocol Π for ME_k, $\delta(\Pi) > 0$ as the input $(0, \ldots, 0)$ is obtained in case $B = 1$ with probability 2^{-k}.

The next proposition lower bounds the information that the transcript $\Pi(X)$ reveals about the input X to an external observer, as a function of the error probability $\delta(\Pi)$.

Proposition 7 (Information lower bound for ME_k)**.** Let Π be a private-coin protocol for the ME_k problem. Then,

$$I(X; \Pi(X) \mid B = 1) \geq \frac{1}{2}(1 - 2\sqrt{\delta(\Pi) + 1/(2^{k+1} - 2)}) \geq \frac{1}{2}(1 - 2\sqrt{\delta(\Pi) + 1/6}).$$

Proof. Denote $\vec{1}_k = (1, \ldots, 1) \in \{0,1\}^k$ and $\vec{0}_k = (0, \ldots, 0) \in \{0,1\}^k$. For $x \in \{0,1\}^k$, denote $\bar{x} = x \oplus \vec{1}_k$.

Partition all 2^k possible inputs $x = (x^1, \ldots, x^k) \in \{0,1\}^k$ into 2^{k-1} ordered pairs of the form $(x, \bar{x})$, where $x^1 = 0$. Define the random variable W, distributed as a uniform pair $(X, \bar{X})$. For a pair $w = (x, \bar{x})$, denote $w_0 = x$ and $w_1 = \bar{x}$.

Since the input X determines W, then

$$\begin{aligned} I(X;\Pi(X) \mid B=1) &= I(W,X;\Pi(X) \mid B=1) \\ &= I(W;\Pi(X) \mid B=1) + I(X;\Pi(X) \mid W,B=1) \\ &\geq I(X;\Pi(X) \mid W,B=1) \\ &= \mathop{\mathrm{E}}_{w}[I(X;\Pi(X) \mid W=w,B=1)], \end{aligned} \tag{1}$$

where the second equality is by the chain rule of mutual information.

Given $W=w$ and assuming $B=1$, we have $X \in_R \{w_0,w_1\}$ with uniform probability. Hence, by Proposition 3, we obtain for each w

$$I(X;\Pi(X) \mid W=w,B=1) \geq h^2(\Pi_{w_0},\Pi_{w_1}). \tag{2}$$

Combining (1) and (2), by an averaging argument over w, there exists a specific pair of inputs $(x',\bar{x'})$ such that

$$I(X;\Pi(X) \mid B=1) \geq h^2(\Pi_{x'},\Pi_{\bar{x'}}).$$

Moreover, by Proposition 5,

$$h^2(\Pi_{x'},\Pi_{\bar{x'}}) = h^2(\Pi_{\vec{0}_k},\Pi_{\vec{1}_k}).$$

Combining, we obtain

$$I(X;\Pi(X) \mid B=1) \geq h^2(\Pi_{\vec{0}_k},\Pi_{\vec{1}_k}). \tag{3}$$

Remark 6. If $\Pi(\vec{1}_k)$ errs with small probability (close to $\delta_1(\Pi)$), then we can use Proposition 4 to complete the proof. However, since we only have an average-case guarantee about Π, its behaviour on $\vec{1}_k$ is undetermined. Thus, we continue by finding and analyzing a non-zero input on which Π errs with small probability.

For any $x \in \{0,1\}^k$, by Proposition 6,

$$h^2(\Pi_{\vec{0}_k},\Pi_x) + h^2(\Pi_{\bar{x}},\Pi_{\vec{1}_k}) \leq 2h^2(\Pi_{\vec{0}_k},\Pi_{\vec{1}_k}).$$

Therefore, for any $x \in \{0,1\}^k$,

$$h^2(\Pi_{\vec{0}_k},\Pi_{\vec{1}_k}) \geq \frac{1}{2}h^2(\Pi_{\vec{0}_k},\Pi_x). \tag{4}$$

Combining (3) and (4), we deduce that for any input $x \in \{0,1\}^k$,

$$I(X;\Pi(X) \mid B=1) \geq h^2(\Pi_{\vec{0}_k},\Pi_{\vec{1}_k}) \geq \frac{1}{2}h^2(\Pi_{\vec{0}_k},\Pi_x). \tag{5}$$

Specifically, let $x^* \neq \vec{0}_k$ be an input such that

$$\Pr[\Pi_{\text{out}}(\Pi(x^*)) = 0] \leq \delta_1(\Pi) + 1/(2^k-1)$$

(such an input exists by an averaging argument over all $x \neq \vec{0}_k$). On the other hand, $\Pr[\Pi_{\text{out}}(\Pi(\vec{0}_k)) = 0] \geq 1 - \delta_0(\Pi)$. Consequently, combining (5) for x^* with Proposition 4, we obtain

$$\begin{aligned} I(X;\Pi(X) \mid B=1) &\geq \frac{1}{2} h^2(\Pi_{\vec{0}_k}, \Pi_{x^*}) \\ &\geq \frac{1}{2}(1 - \sqrt{2(\delta_0(\Pi) + \delta_1(\Pi) + 1/(2^k - 1))}) \\ &= \frac{1}{2}(1 - 2\sqrt{\delta(\Pi) + 1/(2^{k+1} - 2)}) \\ &\geq \frac{1}{2}(1 - 2\sqrt{\delta(\Pi) + 1/6}), \end{aligned}$$

where the final inequality follows since $k \geq 2$. ■

3.2 The Multi-Dimensional Missing-Element Problem

The general (multi-dimensional) multiparty Missing-Element communication problem ($\text{ME}_{k,n}$) is a problem with k players which is a generalization of the unit Missing-Element problem to n dimensions (namely, $\text{ME}_k \equiv \text{ME}_{k,1}$). The input to the problem is denoted $X = (X^1, \ldots, X^k) \in \{0,1\}^{k \cdot n}$, where each player $i \in [k]$ gets a vector $X^i \in \{0,1\}^n$ whose distribution is defined as follows. At the beginning of the game a bit $B \in \{0,1\}$ is chosen uniformly at random and an index $j \in [n]$ is chosen independently and uniformly at random (yet, the specific distribution in which j is chosen will not be a factor in our analysis, and we will mostly not treat it as a random variable). B and j determine the input distribution as follows.

> If $B = 0$, then $X^i_j = 0$ for each $i \in [k]$ (i.e., $X_j = (X^1_j, \ldots, X^k_j) = (0, \ldots, 0)$), while for all $\ell \neq j$, X^i_ℓ is chosen independently uniformly at random.
>
> If $B = 1$, each $X^i \in \{0,1\}^n$ is chosen independently uniformly at random.

Let Π be a protocol for the $\text{ME}_{k,n}$ problem. We define the following types of error probabilities.

> For $b \in \{0,1\}$ and $j \in [n]$, let $\delta_{b,j}(\Pi) = \Pr[\Pi_{\text{out}}(\Pi(X)) \neq b \mid B = b, J = j]$.
>
> The error probability of Π on index $j \in [n]$ is $\delta_j(\Pi) = \text{E}_B[\delta_{B,j}(\Pi)] = (\delta_{0,j}(\Pi) + \delta_{1,j}(\Pi))/2$.
>
> The overall error probability of Π is $\delta(\Pi) = \max_{j \in [n]}\{\delta_j(\Pi)\}$.
>
> For $b \in \{0,1\}$, we further define $\delta_{b,*}(\Pi) = \max_{j \in [n]}\{\delta_{b,j}(\Pi)\}$.

The next proposition lower bounds the information that the transcript $\Pi(X)$ in the $\text{ME}_{k,n}$ problem reveals about the input X to an external observer, as a function of the error probability $\delta(\Pi)$. It uses a direct sum technique for information complexity [CSWY01, BJKS04].

Proposition 8 (Information lower bound for $\text{ME}_{k,n}$)**.** Let Π be a private-coin protocol for the $\text{ME}_{k,n}$ problem. Then,

$$I(X;\Pi(X) \mid B=1) \geq \frac{n}{2} \cdot (1 - 2\sqrt{\delta(\Pi) + 1/6}).$$

Proof. We have

$$\begin{aligned}
I(X;\Pi(X) \mid B=1) &= \sum_{j=1}^{n} I(X_j;\Pi(X) \mid B=1, X_{<j}) \\
&= \sum_{j=1}^{n} (H(X_j \mid B=1, X_{<j}) - H(X_j \mid \Pi(X), B=1, X_{<j})) \\
&= \sum_{j=1}^{n} (H(X_j \mid B=1) - H(X_j \mid \Pi(X), B=1, X_{<j})) \\
&\geq \sum_{j=1}^{n} (H(X_j \mid B=1) - H(X_j \mid \Pi(X), B=1)) \\
&= \sum_{j=1}^{n} I(X_j;\Pi(X) \mid B=1),
\end{aligned}$$

where the first equality is by the chain rule for mutual information, the third equality uses the fact that given $B=1$, X_j and $X_{<j}$ are independent, and the inequality holds since conditioning cannot increase entropy. Therefore, by an averaging argument there exists $j \in [n]$ such that

$$I(X_j;\Pi(X) \mid B=1) \leq \tfrac{1}{n} I(X;\Pi(X) \mid B=1). \tag{6}$$

Fix such j. Given Π we design a protocol Π' for the (unit) ME_k problem as follows. On input $Y \in \{0,1\}^k$ (where player $i \in [k]$ is given Y^i), each player $i \in [k]$ sets $X_j^i = Y^i$ (i.e., we have $X_j = Y$) and picks $X_{-j}^i \in \{0,1\}^{n-1}$ uniformly at random using private coin tosses (where $X_{-j}^i = (X_1^i, \ldots, X_{j-1}^i, X_{j+1}^i, X_n^i)$). The players then run $\Pi(X)$ on $X \in \{0,1\}^{k \cdot n}$.

Note that the players perfectly simulate the input distribution of the $\mathrm{ME}_{k,n}$ problem with $B = B'$, and hence $\delta(\Pi') = \delta_j(\Pi)$. Moreover $\Pi'(Y) = \Pi(X)$. Therefore,

$$\begin{aligned}
I(X_j;\Pi(X) \mid B=1) &= I(Y;\Pi'(Y) \mid B'=1) \\
&\geq \frac{1}{2}(1 - 2\sqrt{\delta(\Pi') + 1/6}) \\
&= \frac{1}{2}(1 - 2\sqrt{\delta_j(\Pi) + 1/6}) \\
&\geq \frac{1}{2}(1 - 2\sqrt{\delta(\Pi) + 1/6}),
\end{aligned}$$

where the first inequality is by Proposition 7. Combining with (6), we obtain

$$I(X;\Pi(X) \mid B=1) \geq \frac{n}{2}(1 - 2\sqrt{\delta(\Pi) + 1/6}).$$

■

Proposition 9 (Private-coin to public-coin reduction)**.** If there is a public-coin protocol Π for the $\mathrm{ME}_{k,n}$ problem with $\delta(\Pi) = \delta$, then there is a private-coin protocol Π' for the problem with $\delta(\Pi') \leq 2\delta^{1/2}$ and communication cost at most $\mathrm{C}_v(\Pi) + \lceil \log n \rceil$.

Proof. For a string r and $j \in [n]$, denote by $\delta_j^r(\Pi)$ the value of $\delta_j(\Pi)$ conditioned on r being the public random string of Π. Let R denote a random variable for the public string of Π. For every $j \in [n]$,

$$\delta \geq \delta_j(\Pi) = \sum_r \Pr[R = r] \cdot \delta_j^r(\Pi) = \mathop{\mathrm{E}}_R[\delta_j^R(\Pi)],$$

hence $\mathrm{E}_{J,R}[\delta_J^R(\Pi)] \leq \delta$. Therefore, there exists a public string r' such that $\mathrm{E}_J[\delta_J^{r'}(\Pi)] \leq \delta$. By Markov's inequality,

$$\Pr_J[\delta_J^{r'}(\Pi) > \delta^{-1/2} \cdot \delta] \leq \delta^{1/2}.$$

The private-coin protocol Π' has r' embedded. The first player that communicates in Π' initially draws $V \in [n]$ uniformly at random and writes it on the shared blackboard. Then, the players run Π with the public string r', where each player $i \in [k]$ rotates the input X^i by V positions, namely, defines the new input to Π' as $Y_\ell^i = X_{\ell+V \bmod n}^i$ for all $\ell \in [n]$.

Since V can be encoded using $\lceil \log n \rceil$ bits, the communication cost of Π' is at most $\mathrm{C}_v(\Pi) + \lceil \log n \rceil$ as claimed. Due to the randomization of the input, $\delta_j(\Pi')$ is independent of j for every $j \in [n]$ and bounded as

$$\delta_j(\Pi') \leq \delta^{1/2} \cdot \Pr_V[\delta_{j+V \bmod n}^{r'}(\Pi) \leq \delta^{1/2}] + 1 \cdot \Pr_V[\delta_{j+V \bmod n}^{r'}(\Pi) > \delta^{1/2}] \leq \delta^{1/2} + \delta^{1/2} = 2\delta^{1/2}.$$

Hence $\delta(\Pi') \leq 2\delta^{1/2}$. ■

Proposition 10 ($\mathrm{C}_v(\Pi)$ lower bound for small constant error)**.** Any public-coin protocol Π for the $\mathrm{ME}_{k,n}$ problem with $\delta(\Pi) = \delta$ satisfies $\mathrm{C}_v(\Pi) \geq \frac{n}{2} \cdot (1 - 2\sqrt{2\delta^{1/2} + 1/6}) - \lceil \log n \rceil$.

Proof. Let Π be a public-coin protocol for the $\mathrm{ME}_{k,n}$ problem with $\delta(\Pi) = \delta$. By Proposition 9, there is a private-coin protocol Π' for the $\mathrm{ME}_{k,n}$ problem with $\delta(\Pi') \leq 2\delta^{1/2}$ and communication cost C' such that $\mathrm{C}' \leq \mathrm{C}_v(\Pi) + \lceil \log n \rceil$.

By Proposition 8,

$$I(X; \Pi'(X) \mid B = 1) \geq \frac{n}{2} \cdot (1 - 2\sqrt{\delta(\Pi') + 1/6}) \geq \frac{n}{2} \cdot (1 - 2\sqrt{2\delta^{1/2} + 1/6}).$$

Since

$$H(\Pi'(X) \mid B = 1) \geq I(X; \Pi'(X) \mid B = 1),$$

there exists an input x (in case $B = 1$) for which the length of $\Pi'(x)$ (for some choice of the private coins of the players) is at least $H(\Pi'(X) \mid B = 1) \geq \frac{n}{2} \cdot (1 - 2\sqrt{2\delta^{1/2} + 1/6})$. Hence, $\frac{n}{2} \cdot (1 - 2\sqrt{2\delta^{1/2} + 1/6}) \leq \mathrm{C}' \leq \mathrm{C}_v(\Pi) + \lceil \log n \rceil$. Therefore, $\mathrm{C}_v(\Pi) \geq \frac{n}{2} \cdot (1 - 2\sqrt{2\delta^{1/2} + 1/6}) - \lceil \log n \rceil$.

■

Proposition 11 (Amplifying constant success probability)**.** Assume there is a public-coin protocol Π for the $\mathrm{ME}_{k,n}$ problem with constant $\delta(\Pi) = \delta < 1/2$. Then, for any constant $\delta' > 0$, there is a public-coin protocol Π' for the $\mathrm{ME}_{k',n}$ problem where $k' = O(k)$ such that $\delta(\Pi') \leq \delta'$ and $\mathrm{C}_v(\Pi') \leq O(\mathrm{C}_v(\Pi))$.

Proof. For an integer parameter $t > 0$, denote $k' = t \cdot k$. The protocol Π' partitions the k' players into t groups of size k. It runs Π independently on each group after (independently) randomizing the input for each group by shifting the input. Specifically, in each group of size k, before running Π, each player i applies an independent uniform shift V to the input by defining $Y^i_\ell = X^i_{\ell+V \bmod n}$ for all $\ell \in [n]$. The shift of each group is specified in the common shared random string. Next, we describe how the output of Π' is defined according to the outcomes of all t executions.

Denote

$$\alpha_0 = \frac{1}{n}\sum_{j=1}^{n} \Pr[\Pi_{\text{out}}(\Pi(X)) = 1 \mid B = 0, J = j], \text{ and}$$
$$\alpha_1 = \frac{1}{n}\sum_{j=1}^{n} \Pr[\Pi_{\text{out}}(\Pi(X)) = 1 \mid B = 1, J = j].$$

Let $\epsilon = 1/2 - \delta$. For every $j \in [n]$ we have

$$\begin{aligned}&\Pr[\Pi_{\text{out}}(\Pi(X)) = 1 \mid B = 1, J = j] - \Pr[\Pi_{\text{out}}(\Pi(X)) = 1 \mid B = 0, J = j] \\ =&\, 1 - \delta_{1,j}(\Pi) - \delta_{0,j}(\Pi) = 1 - 2\delta_j(\Pi) \geq 1 - 2\delta = 2\epsilon.\end{aligned}$$

Hence $\alpha_1 - \alpha_0 \geq 2\epsilon$. Note that $\alpha_0, \alpha_1, \epsilon$ are fixed and known values.

Denote by $Z_1, \ldots, Z_t \in \{0, 1\}$ random variables for the outputs of the t executions of Π, and let $M = \sum_{i=1}^{t} Z_i$. The output of Π' is defined to be 0 if $M \leq t(\alpha_0 + \epsilon)$ and 1 otherwise.

Since the number of players in Π' is $k' = t \cdot k$ and $\mathrm{C}_v(\Pi') \leq t \cdot \mathrm{C}_v(\Pi)$, it remains to prove that we can choose t as a function of the constants ϵ and δ' so $\delta(\Pi') \leq \delta'$.

In case $B = 0$, since $j+V \bmod n$ is uniformly distributed in each execution, we have for every $i \in [t]$, $\Pr[Z_i] = \alpha_0$ (hence $\mathrm{E}[M] = t\alpha_0$), while the variables $Z_1, \ldots, Z_t$ are independent. Consequently, by Proposition 1, for every $j \in [n]$,

$$\delta_{0,j}(\Pi') = \Pr[M - t\alpha_0 > t\epsilon \mid B = 0] \leq \exp(-2t^2 \cdot \epsilon^2/t) = \exp(-2t \cdot \epsilon^2).$$

Choosing $t \geq \ln(1/\delta')/(2\epsilon^2)$ suffices to have $\delta_{0,j}(\Pi') \leq \delta'$. The analysis for $\delta_{1,j}(\Pi')$ is similar, noting that in case $B = 1$, $\Pr[Z_i] = \alpha_1 \geq \alpha_0 + 2\epsilon$ (hence $\mathrm{E}[M] \geq t(\alpha_0 + 2\epsilon)$). We conclude that we can indeed choose t as required. ■

Proposition 12 ($\mathrm{C}_v(\Pi)$ lower bound for constant error)**.** Any public-coin protocol Π for the $\mathrm{ME}_{k,n}$ problem with constant $\delta(\Pi) = \delta < 1/2$ has $\mathrm{C}_v(\Pi) \geq \Omega(n)$.

Proof. Let Π be a public-coin protocol for the $\mathrm{ME}_{k,n}$ problem with $\delta(\Pi) = \delta$. By proposition 11, for any $\delta' > 0$, there is a public-coin protocol Π' for the $\mathrm{ME}_{k',n}$ problem for $k' = O(k)$ with $\delta(\Pi') \leq \delta'$ and $\mathrm{C}_v(\Pi') \leq O(\mathrm{C}_v(\Pi))$. Choosing δ' sufficiently small (so that $2\sqrt{2(\delta')^{1/2}} + 1/6 < 1$), any applying proposition 10 to Π', we conclude that $\mathrm{C}_v(\Pi') \geq \frac{n}{2} \cdot (1 - 2\sqrt{2(\delta')^{1/2}} + 1/6) - \lceil \log n \rceil \geq \Omega(n)$. Hence $\mathrm{C}_v(\Pi) \geq \Omega(n)$, as claimed. ■

We now prove the main result about the $\mathrm{ME}_{k,n}$ problem.

Proposition 13 ($\mathrm{C}_v(\Pi)$ lower bound for error approaching 1/2)**.** Let $\gamma = \gamma(n)$ be a function such that $0 < \gamma(n) < 1/2$ for all sufficiently large n. Assume that there is public-coin protocol Π for the $\mathrm{ME}_{k,n}$ problem such that for sufficiently large n,

$$\delta_{0,*}(\Pi) \leq 1 - \gamma \text{ and } \delta_{1,*}(\Pi) \leq \gamma/4.$$

Then, $\mathrm{C}_v(\Pi) \geq \gamma \cdot \Omega(n)$.

Proof. Given Π, we construct below a public-coin protocol Π' for the $\mathrm{ME}_{k',n}$ problem with $k' = \lfloor \gamma^{-1} \rfloor \cdot k$, $\mathrm{C}_v(\Pi') \leq \lfloor \gamma^{-1} \rfloor \cdot \mathrm{C}_v(\Pi)$, and $\delta(\Pi') \leq 0.49$ (for n sufficiently large). Therefore, by Proposition 12, $\mathrm{C}_v(\Pi') \geq \Omega(n)$, hence $\mathrm{C}_v(\Pi) \geq \gamma \cdot \mathrm{C}_v(\Pi') \geq \gamma \cdot \Omega(n)$, as claimed.

Similarly to the protocol in the proof of Proposition 11, Π' partitions the players into $\lfloor \gamma^{-1} \rfloor$ groups, where each group independently runs Π. The only difference from the proof of Proposition 11 is in the computation of the output. In this case, the output of Π' is defined to be 1 if all $\lfloor \gamma^{-1} \rfloor$ executions of Π output 1, and 0 otherwise.

By a union bound over the $\lfloor \gamma^{-1} \rfloor$ executions,

$$\delta_1(\Pi') \leq \lfloor \gamma^{-1} \rfloor \cdot \gamma/4 \leq 1/4.$$

On the other hand, in case $B = 0$, Π' errs only if all the $\lfloor \gamma^{-1} \rfloor$ independent executions of Π err. Therefore, for sufficiently large n,

$$\delta_0(\Pi') \leq (1-\gamma)^{\lfloor \gamma^{-1} \rfloor} \leq e^{-1}/(1-\gamma) \leq 2e^{-1},$$

since $\gamma < 1/2$. Thus, we indeed have $\delta(\Pi') \leq (1/4 + 2e^{-1})/2 < 0.49$. ■

4 Time-Space Lower Bounds for Computing Functions with in the Random-Query Model

In this section we use Proposition 13 to prove time-space tradeoff lower bounds under the random-query model with the independent distribution in the bounded-error regime.

Proposition 14 (Reduction from $\mathrm{ME}_{k,n}$ to computing f in the random-query model)**.** Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function such that there is a branching program of length $\mathrm{T} < n^2/4$ and width 2^{S} that computes f with error $\delta < 1/2$ for sufficiently large n under the random-query model with the independent distribution. Then, for sufficiently large n, there is public-coin protocol Π for the $\mathrm{ME}_{k,n}$ problem with $k = \lceil 4\mathrm{T}/n \rceil$, $\mathrm{C}_v(\Pi) \leq \mathrm{S} \cdot (k-1) + 1 = \mathrm{S} \cdot (\lceil 4\mathrm{T}/n \rceil - 1) + 1$, and

$$\delta_{0,*}(\Pi) \leq 1 + 2^{-n/20} - \frac{1}{2n}\mathrm{I}[f] \text{ and } \delta_{1,*}(\Pi) \leq \delta + 2^{-n/20}.$$

Proof.

The protocol. Let $\mathcal{P}$ be the branching program assumed in the propsition. We construct a public-coin protocol Π with the desired parameters. In order to avoid confusion between the input to the branching program $\mathcal{P}$ and the input to the protocol Π, throughout the proof we use

somewhat different terminology and notation for $\mathcal{P}$ than described in Section 2.2. Specifically, the input indices to $\mathcal{P}$ are referred to as "queries", while the input bits are referred to as "answers".

Denote $m = \lceil n/4 \rceil$. First, we describe how each player $i \in [k]$ generates m query-answer pairs to $\mathcal{P}$, denoted by $(Q_1^i, A_1^i), \ldots, (Q_m^i, A_m^i)$. Assuming that the queries $Q_1^i, \ldots, Q_m^i$ have been generated (as described below), the corresponding answers $A_1^i, \ldots, A_m^i$ are generated using the first n bits of the random public string R by defining $A_\ell^i = R_{Q_\ell^i}$ for each $\ell \in [m]$. Thus, the input to $\mathcal{P}$ (the answers to the queries) is a uniform n-bit string.

The queries $Q_1^i, \ldots, Q_m^i$ are generated as follows. On input $X^i \in \{0,1\}^n$, player i first verifies that the Hamming weight of X^i is at least m (otherwise, the player defines the queries arbitrarily and we assume the protocol fails). Then, the input is treated as a subset of $[n]$ of size at least m. The player uniformly permutes (reorders) the elements of this set (using private coins) and considers the sequence of the first m elements, denoted $(E_1, E_2, \ldots, E_m)$, where $E_\ell \in [n]$ for $\ell \in [m]$. Then, the player shifts each element by $V \in [n]$, where V is an independent uniform public value that is common to all players, defined in the public random string. Concretely, the player defines $(G_1, \ldots, G_m) = (E_1 + V \bmod n, \ldots, E_m + V \bmod n)$.

Assume that $(Q_1^i, \ldots, Q_\ell^i)$ have already been generated. We descrine how player i generates $Q_{\ell+1}^i$. Note that $(G_1, \ldots, G_m)$ (and $(E_1, \ldots, E_m)$) are selected from $[n]$ without replacement, but the queries in the random-query model need to be drawn with replacement. Specifically, with probability $\frac{\ell}{n}$, we set $Q_{\ell+1}^i = Q_{\ell'}^i$ where $\ell' \in [\ell]$ is chosen uniformly at random. Otherwise (with probability $1 - \frac{\ell}{n}$), $Q_{\ell+1}^i$ is defined to be the next unused element in the sequence $(G_1, \ldots, G_m)$.

The protocol Π begins by the first player feeding the query-answer pairs $(Q_1^1, A_1^1), \ldots, (Q_m^1, A_m^1)$ to $\mathcal{P}$ and writing an encoding of the vertex reached on the blackboard. The second player continues the execution of $\mathcal{P}$ from the previous state using $(Q_1^2, A_1^2), \ldots, (Q_m^2, A_m^2)$ and so forth.

Finally, the last player finishes the execution after feeding $\mathcal{P}$ with at most m query-answer pairs (as indeed $k \cdot m = \lceil 4\mathrm{T}/n \rceil \cdot \lceil n/4 \rceil \geq \mathrm{T}$), and computes the output of $\mathcal{P}$ which we denote by $W \in \{0,1\}$. The last player then computes the true value of f on the input to $\mathcal{P}$, $W' = f(R_1, \ldots, R_n)$, and writes the output of Π on the blackboard, defined as 1 if $W = W'$, and 0 otherwise.

Analysis. Since the width of $\mathcal{P}$ is 2^{S}, each of the (at most) $k-1$ vertex encodings written on the blackboard by the first $k-1$ players has length of S bits. Including the final bit written by the last player, we indeed have $\mathrm{C}_v(\Pi) \leq \mathrm{S} \cdot (k-1) + 1$.

We now analyze the error probability of Π. We denote by $\mathcal{E}_1$ the event that all players have inputs with Hamming weight at least m. Recall that the input of each player is an n-bit Boolean vector which is uniformly distributed in case $B = 1$, or has $n-1$ uniformly distributed entries in case $B = 0$.

By a standard Chernoff bound, the probability that the Hamming weight of each player's input is less than $m = \lceil n/4 \rceil$ is most $2^{-n/10}$ for sufficiently large n (even if $B = 0$). By a union bound over the $k < n$ players,

$$\Pr[\neg \mathcal{E}_1] \leq k \cdot 2^{-n/10} < n \cdot 2^{-n/10} < 2^{-n/20}$$

for sufficiently large n, independently of B.

If $B = 1$, conditioned on $\mathcal{E}_1$, the T queries of Π are uniform and independent elements of $[n]$, as in the random-query model with the independent distribution. Hence, by the properties of $\mathcal{P}$, $W \neq W'$ (and Π errs) with probability δ, implying $\delta_{1,*}(\Pi) \leq \delta + 2^{-n/20}$, as claimed.

If $B = 0$, there is a missing index $j \in [n]$. We upper bound $\delta_{0,*}(\Pi) = \Pr[W = W' \mid B = 0]$. This is done by conditioning on two events. The first event is $\mathcal{E}_1$, defined above. The second event (denoted $\mathcal{E}_2$) is that the shifted missing index $J' = j + V \bmod n$ is pivotal for f on input $(R_1, \ldots, R_n)$ (the answers given to $\mathcal{P}$), and thus $W' = f(R_1, \ldots, R_n) \neq f(R_1, \ldots, R_{J'-1}, R_{J'} \oplus 1, R_{J'+1}, \ldots, R_n)$.

By Proposition 2,

$$\Pr[\mathcal{E}_2 \mid B = 0] = \Pr_{J', R_1, \ldots, R_n}[J' \text{ is pivotal for } f \text{ on } (R_1, \ldots, R_n)] = \frac{1}{n}\mathrm{I}[f].$$

In addition,

$$\Pr[W = W' \mid \mathcal{E}_1 \wedge \mathcal{E}_2, B = 0] = 1/2,$$

where the probability is taken over the uniform choice of $R_{J'}$ (which is never given to $\mathcal{P}$ as an answer). Combining these calculations, for sufficiently large n,

$$\begin{aligned}
\delta_{0,*}(\Pi) &= \Pr[W = W' \mid B = 0] \\
&\leq \Pr[W = W' \mid \mathcal{E}_1 \wedge \mathcal{E}_2, B = 0] \cdot \Pr[\mathcal{E}_1 \wedge \mathcal{E}_2 \mid B = 0] + \Pr[\neg\mathcal{E}_1 \vee \neg\mathcal{E}_2 \mid B = 0] \\
&\leq \frac{1}{2} \cdot \Pr[\mathcal{E}_2 \mid B = 0] + \Pr[\neg\mathcal{E}_1 \mid B = 0] + \Pr[\neg\mathcal{E}_2 \mid B = 0] \\
&\leq 1 + 2^{-n/20} - \frac{1}{2n}\mathrm{I}[f].
\end{aligned}$$

■

Remark 7. In the reduction, rather than limiting each player to generating $\lceil n/4 \rceil$ query-answer pairs and feeding them to $\mathcal{P}$, we could allow the players to fully use their inputs for this purpose. However, this has a (limited) penalty in communication, since in this alternative reduction each player needs to write on the blackboard the number of levels of $\mathcal{P}$ it has executed, so the next player can continue the execution from the correct level.

Proposition 15 (Amplifying success probability for branching programs)**.** Assume that there is a branching program of length T and width 2^{S} that computes $f : \{0,1\}^n \to \{0,1\}$ with constant error $\delta < 1/2$ under the random-query model with the independent distribution. Let $\delta' = \delta'(n) > 0$. Then, there is a branching program of length $O(\log(1/\delta')) \cdot \mathrm{T}$ and width $O(\log(1/\delta')) \cdot 2^{\mathrm{S}}$ that computes f with error δ' under the random-query model with the independent distribution.

Proof. Denote by $\mathcal{P}$ the assumed branching program. We construct a branching program $\mathcal{P}'$ with the desired parameters as follows. Let $t = t(n)$ be an odd integer parameter to be chosen later. $\mathcal{P}'$ runs t independent executions of $\mathcal{P}$ sequentially, and outputs a majority vote on the t outputs.

Calculating the majority votes across the t outputs (with $2t + 1$ possible outcomes) requires increasing the width by a multiplicative factor of at most $2t + 1$. The length of $\mathcal{P}'$ is $t \cdot \mathrm{T}$. It remains to show that we can choose $t \leq O(\log(1/\delta'))$ to obtain the desired error probability δ'.

Denote the input to $\mathcal{P}'$ by x. Let $Z_1, \ldots, Z_t \in \{0,1\}$ be (independent) random variables such that $Z_i = 1$ if the output of the i'th execution of $\mathcal{P}$ is incorrect, namely $Z_i \neq f(x)$. By the properties

of $\mathcal{P}$, for any $i \in [t]$, $\Pr[Z_i = 1] \leq \delta$. Denote $\Pr[Z_i = 1] = \alpha \leq \delta < 1/2$. Let $M = \sum_{i=1}^{t} Z_i$. Then $\mathrm{E}[M] = t \cdot \alpha$. By Proposition 1,

$$\begin{aligned}\Pr[\mathcal{P}' \text{ errs}] &\leq \Pr[M > t/2] \\ &= \Pr[M - t \cdot \alpha > t(1/2 - \alpha)] \leq \exp(-2t(1/2-\alpha)^2) \leq \exp(-2t(1/2-\delta)^2).\end{aligned}$$

Therefore, choosing $t \geq \frac{\ln(1/\delta')}{2(1/2-\delta)^2} = \Theta(\log(1/\delta'))$ suffices. ■

We can now prove the formal version of Theorem 1.

Theorem 2 (Time-space tradeoff for computing f in the random-query model)**.** Assume that there is a branching program of length T and width 2^{S} that computes $f : \{0,1\}^n \to \{0,1\}$ with constant error $\delta < 1/2$ under the random-query model with the independent distribution. Then,

$$\mathrm{T} \cdot (\mathrm{S} + \log\log(\tfrac{9n}{\mathrm{I}[f]})) \cdot \log(\tfrac{9n}{\mathrm{I}[f]}) \geq \Omega(n) \cdot \mathrm{I}[f].$$

In particular,

$$\mathrm{T} \cdot (\mathrm{S} + \log\log n) \geq \Omega(n/\log n) \cdot \mathrm{I}[f],$$

and if $\mathrm{I}[f] \geq \Omega(n)$, then

$$\mathrm{T} \cdot \mathrm{S} \geq \Omega(n^2).$$

Proof. We assume that $\mathrm{I}[f] \geq \Omega(1/n)$, as otherwise there is nothing to prove.

Denote by $\mathcal{P}$ the assumed branching program. Define $\beta = \beta(n) = \frac{1}{9n}\mathrm{I}[f]$. By Proposition 15 there is a branching program $\mathcal{P}'$ of length $\mathrm{T}' \leq O(\log(1/\beta)) \cdot \mathrm{T}$ and width $2^{\mathrm{S}'} \leq O(\log(1/\beta)) \cdot 2^{\mathrm{S}}$ that computes f with error β under the random-query model with the independent distribution.

If $\mathrm{T}' \geq n^2/4$, then the claim holds trivially. Hence, assume that $\mathrm{T}' < n^2/4$. Then, by Proposition 14, for sufficiently large n, there is public-coin protocol Π for the $\mathrm{ME}_{k,n}$ problem with $k = \lceil 4\mathrm{T}'/n \rceil$, $\mathrm{C}_v(\Pi) \leq \mathrm{S}' \cdot (\lceil 4\mathrm{T}'/n \rceil - 1) + 1$, and

$$\delta_{0,*}(\Pi) \leq 1 + 2^{-n/20} - \frac{1}{2n}\mathrm{I}[f] \text{ and } \delta_{1,*}(\Pi) \leq \beta + 2^{-n/20}.$$

We would now like to apply Proposition 13 to Π. For this purpose, define $\gamma = \gamma(n) = \frac{1}{2n}\mathrm{I}[f] - 2^{-n/20}$, thus $\delta_{0,*}(\Pi) \leq 1 - \gamma$. Moreover, for sufficiently large n (recalling that $\mathrm{I}[f] \geq \Omega(1/n)$),

$$\delta_{1,*}(\Pi) \leq \beta + 2^{-n/20} = \frac{1}{9n}\mathrm{I}[f] + 2^{-n/20} \leq (\frac{1}{2n}\mathrm{I}[f] - 2^{-n/20})/4 = \gamma/4.$$

Hence, we can indeed apply Proposition 13 to Π and conclude that

$$\mathrm{S}' \cdot (\lceil 4\mathrm{T}'/n \rceil - 1) + 1 \geq \mathrm{C}_v(\Pi) \geq \gamma \cdot \Omega(n),$$

or $\mathrm{S}' \cdot \mathrm{T}' \geq \Omega(n) \cdot \mathrm{I}[f]$. Substituting S' and T' we obtain

$$\mathrm{T} \cdot (\mathrm{S} + \log\log(\tfrac{9n}{\mathrm{I}[f]})) \cdot \log(\tfrac{9n}{\mathrm{I}[f]}) \geq \Omega(n) \cdot \mathrm{I}[f].$$

■

Theorem 3 (Time-space tradeoff for computing Majority in the random-query model)**.** Assume that there is a branching program of length T and width 2^S that computes Majority$(x_1, \ldots, x_n)$ with constant error $\delta < 1/2$ under the random-query model with the independent distribution. Then,

$$T \cdot S \geq \Omega(n^2).$$

Proof. The proof is similar to that of Theorem 2, but with $I[f]$ replaced with $n/2$. This is possible as we will show how to replace the bound on the error probabilities in Proposition 14 by a stronger bound for $\delta_{0,*}(\Pi)$, specifically,

$$\delta_{0,*}(\Pi) \leq 3/4 + 2^{-n/20} \text{ and } \delta_{1,*}(\Pi) \leq \beta + 2^{-n/20},$$

where now we choose $\beta = \frac{1}{9n} \cdot n/2 = 1/18$. Plugging these bounds into the proof of Theorem 2 gives the claimed result. It remains to modify the reduction protocol of Proposition 14 and analyze the error probabilities.

We assume that n is odd, so the majority is well-defined. The only difference in the reduction protocol is in way that the first n bits of the random public string R are chosen. Instead of picking them uniformly at random, they are picked by first choosing $U \in \{0, 1\}$ uniformly. If $U = 0$, then $(R_1, \ldots, R_n)$ is picked as a uniform vector of Hamming weight $(n-1)/2$. If $U = 1$, then $(R_1, \ldots, R_n)$ is picked as a uniform vector of Hamming weight $(n+1)/2$. Next, we redo the error probability analysis.

The analysis in case $B = 1$ remains identical to Proposition 14, hence $\delta_{1,*}(\Pi) \leq \beta + 2^{-n/20}$.

In case $B = 0$, there is a missing index $j \in [n]$. Define the events $\mathcal{E}_1$ and $\mathcal{E}_2$ as in the proof of Proposition 14. The only difference from Proposition 14 is in the analysis of $\mathcal{E}_2$. Recall that $\mathcal{E}_2$ occurs if the shifted missing index $J' = j + V \bmod n$ is pivotal for f on input $(R_1, \ldots, R_n)$, and thus $W' = f(R_1, \ldots, R_n) \neq f(R_1, \ldots, R_{J'-1}, R_{J'} \oplus 1, R_{J'+1}, \ldots, R_n)$.

For Majority we have

$$\Pr[\mathcal{E}_2 \mid B = 0] = \tfrac{n+1}{2} > 1/2$$

regardless of U, since all $(n+1)/2$ bits of $(R_1, \ldots, R_n)$ that agree with the majority are at pivotal indices. In addition,

$$\Pr[W = W' \mid \mathcal{E}_1 \wedge \mathcal{E}_2, B = 0] = 1/2,$$

which follows by the observation that if $\mathcal{E}_1 \wedge \mathcal{E}_2$ occurs then $R_{J'} = U$ is uniformly distributed (and never given to $\mathcal{P}$ as an answer). Combining, for sufficiently large n,

$$\begin{aligned}
\delta_{0,*}(\Pi) &= \Pr[W = W' \mid B = 0] \\
&\leq \Pr[W = W' \mid \mathcal{E}_1 \wedge \mathcal{E}_2, B = 0] \cdot \Pr[\mathcal{E}_1 \wedge \mathcal{E}_2 \mid B = 0] + \Pr[\neg\mathcal{E}_1 \vee \neg\mathcal{E}_2 \mid B = 0] \\
&\leq \frac{1}{2} \cdot \Pr[\mathcal{E}_2 \mid B = 0] + \Pr[\neg\mathcal{E}_1 \mid B = 0] + \Pr[\neg\mathcal{E}_2 \mid B = 0] \\
&\leq 3/4 + 2^{-n/20}.
\end{aligned}$$

■

References

[BEO+13] Mark Braverman, Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. A Tight Bound for Set Disjointness in the Message-Passing Model. In *FOCS 2013*, pages 668–677. IEEE Computer Society, 2013.

[BGY18] Paul Beame, Shayan Oveis Gharan, and Xin Yang. Time-Space Tradeoffs for Learning Finite Functions from Random Evaluations, with Applications to Polynomials. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *COLT 2018*, volume 75 of *Proceedings of Machine Learning Research*, pages 843–856. PMLR, 2018.

[BJKS04] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68(4):702–732, 2004.

[BNS89] László Babai, Noam Nisan, and Mario Szegedy. Multiparty Protocols and Logspace-hard Pseudorandom Sequences (Extended Abstract). In David S. Johnson, editor, *STOC 1989*, pages 1–11. ACM, 1989.

[BSSV03] Paul Beame, Michael E. Saks, Xiaodong Sun, and Erik Vee. Time-space trade-off lower bounds for randomized computation of decision problems. *J. ACM*, 50(2):154–195, 2003.

[CSWY01] Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew Chi-Chih Yao. Informational Complexity and the Direct Sum Problem for Simultaneous Message Complexity. In *FOCS 2001*, pages 270–278. IEEE Computer Society, 2001.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

[Gro09] André Gronemeier. Asymptotically Optimal Lower Bounds on the NIH-Multi-Party Information Complexity of the AND-Function and Disjointness. In Susanne Albers and Jean-Yves Marion, editors, *STACS 2009*, volume 3 of *LIPIcs*, pages 505–516. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 2009.

[GRT19] Sumegha Garg, Ran Raz, and Avishay Tal. Time-Space Lower Bounds for Two-Pass Learning. In Amir Shpilka, editor, *CCC 2019*, volume 137 of *LIPIcs*, pages 22:1–22:39. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[Jay09] T. S. Jayram. Hellinger Strikes Back: A Note on the Multi-party Information Complexity of AND. In Irit Dinur, Klaus Jansen, Joseph Naor, and José D. P. Rolim, editors, *APPROX 2009, and RANDOM 2009*, volume 5687 of *Lecture Notes in Computer Science*, pages 562–573. Springer, 2009.

[MM18] Dana Moshkovitz and Michal Moshkovitz. Entropy Samplers and Strong Generic Lower Bounds For Space Bounded Learning. In Anna R. Karlin, editor, *ITCS 2018*, volume 94 of *LIPIcs*, pages 28:1–28:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[O'D14] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

[Raz19] Ran Raz. Fast Learning Requires Good Memory: A Time-Space Lower Bound for Parity Learning. *J. ACM*, 66(1):3:1–3:18, 2019.

[RZ20] Ran Raz and Wei Zhan. The Random-Query Model and the Memory-Bounded Coupon Collector. In Thomas Vidick, editor, *ITCS 2020*, volume 151 of *LIPIcs*, pages 20:1–20:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[SVW16] Jacob Steinhardt, Gregory Valiant, and Stefan Wager. Memory, Communication, and Statistical Queries. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *COLT 2016*, volume 49 of *JMLR Workshop and Conference Proceedings*, pages 1490–1516. JMLR.org, 2016.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il