Electronic Colloquium on Computational Complexity, Report No. 68 (2025)

Communication Complexity of Equality and Error Correcting Codes

Dale Jacobs[1], John Jeang[1], Vladimir Podolskii[1], Morgan Prior[1], and Ilya Volkovich[2]

[1]Department of Computer Science, Tufts University
[2]Department of Computer Science, Boston College

Abstract

We study the randomized communication complexity of the equality function in the public-coin model. Although the communication complexity of this function is known to be low in the setting where error probability is constant and a large number of random bits are available to players, the complexity grows if the allowed error probability and the amount of randomness are restricted.

We show that randomized protocols for equality and error correcting codes are essentially the same object. That is, given a protocol for equality, we can construct a code, and vice versa.

We substantially extend one of the directions of this connection: any protocol computing a function with a large fooling set can be converted into an error correcting code. As a corollary, we show that among functions with a fooling set of size s, equality on $\log s$ bits has the least randomized communication complexity, regardless of the restrictions on the error probability and the amount of randomness.

Finally, we use the connection to error correcting codes to analyze the randomized communication complexity of equality for varying restrictions on the error probability and the amount of randomness. In most cases, we provide tight bounds. We pinpoint the setting in which tight bounds are still unknown.

1 Introduction

In the standard model of communication complexity, two players, Alice and Bob, are tasked with computing the value of a fixed function $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ on some given input (x, y). However, Alice only knows x and Bob only knows y, and they must communicate to compute $f(x, y)$. Communication complexity studies the amount of communication between players in terms of the number of bits exchanged. In recent decades, communication complexity has become one of the central areas in computational complexity, with numerous applications to other areas, including circuit complexity, proof complexity, online algorithms, and data structures [KN97, RY20, Rou16, Juk12].

The equality function $\mathrm{EQ}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is one of the most fundamental functions in communication complexity. On input (x, y), this function outputs 1 iff $x = y$. This was the first function for which an exponential separation between deterministic and randomized communication complexity was shown (see [Yao79]). Later, EQ played a key role in many results in communication complexity, including direct sum results for the randomized setting [FKNN95], protocols for threshold functions [Nis93], lifting theorems [LM19, dRMN+20], and communication complexity with an equality oracle [CLV19, PSS23].

ISSN 1433-8092

It is well known that the randomized communication complexity of EQ is low for constant error probability. In the private randomness setting, there is an $O(\log n)$ protocol, usually attributed to Rabin and Yao (unpublished, see [KN97, Raz11]). For the case of public randomness, there is an $O(1)$ protocol. The general idea of this protocol is that players can first use their shared randomness to sample a random hash function h. Then Alice sends Bob $h(x)$, and Bob compares it to $h(y)$. The most standard protocol of this type is for players to pick a random vector $r \in \{0,1\}^n$ and compare the inner products $\langle x, r\rangle$ and $\langle y, r\rangle$ over $\mathbb{F}_2$. In the case that x and y are not equal, there is a $1/2$ probability that the compared inner products are also not equal.

Although the randomized complexity of EQ is $O(1)$ for constant error probability and unlimited access to randomness, the complexity can increase when we impose stronger restrictions on these parameters. The error probability for EQ (as well as for any other function) can be reduced to any ε at the cost of a $\log \frac{1}{\varepsilon}$ factor in complexity by a standard error reduction procedure. As for the amount of randomness required, note that the EQ protocol above uses a large number of random bits, which grows as the error probability is reduced. The number of random bits, which we denote by m, can be substantially reduced using the idea from Newman's result on the connection between communication complexity in the public-coin and private-coin models [New91, LMdW21]. Another known construction of randomized protocols for EQ with restricted randomness is through error correcting codes. In such a protocol, players privately encode their inputs via an error correcting code and compare them in a random coordinate (see Section 2.4 or [RY20, Chapter 3]).

As for lower bounds, Canetti and Goldreich [CG93] obtained general lower bounds in the setting with restricted amount of randomness (by reduction to deterministic case, see Fact 4). For equality, their result gives a lower bound of $\Omega\left(\frac{n}{\varepsilon 2^m}\right)$. They also provided examples of matching upper bounds for the case of constant error probability. Moran et al. [MSY16] (see also [LMdW21]) gave a lower bound on communication complexity in the private-coin model in the small error case for functions with large fooling sets. In particular, for equality, their bound implies a lower bound of $\Omega\left(\log n + \log \frac{1}{\varepsilon}\right)$ in the private randomness setting. For public randomness, this implies a lower bound $\Omega\left(\log n + \log \frac{1}{\varepsilon} - m\right)$ (to reduce a protocol with private randomness to one with public randomness one of the players can generate random bits and send them to another player). Despite these results, the entire landscape of communication complexity of EQ for varying ε and m is not fully clear.

Our Results

As mentioned above, error correcting codes give communication protocols for EQ. In our first result, we observe that actually, the converse is also true: communication protocols for EQ give rise to error correcting codes. Thus, error correcting codes and protocols for EQ are essentially the same object.

Main Theorem 1 (Informal version of Theorem 2 and Corollary 3)**.** *Given a protocol for* EQ*, we can construct an error correcting code with equivalent parameters (up to a constant factor), and vice versa. Furthermore, this connection is precise for one-sided error protocols; in this case, protocols and error-correcting codes are in one-to-one correspondence.*

We obtain the following corollary:

Corollary (Informal version of Corollary 4)**.** *Any randomized public-coin protocol for* EQ *can be converted into a one-sided false-positive error protocol with no loss in parameters.*

In particular, any communication protocol for EQ with ε_1 false positive error and ε_0 false negative error can be converted into a one-sided error protocol with $\varepsilon_0 + \varepsilon_1$ false positive error (the communication complexity and the number of random bits used remain the same).

Next, we show that one direction of the result above generalizes to arbitrary functions in the following way:

Main Theorem 2 (Informal version of Theorem 5)**.** *Any randomized public-coin protocol for a function with a large fooling set can be converted into an error correcting code with the same parameters as for* EQ*.*

As a consequence, we obtain the following corollary:

Corollary (Informal version of Corollary 6)**.** *Among all functions with a fooling set of size s,* EQ *on* $\log s$ *bits has the least randomized public-coin communication complexity for all values of error probability and number of random bits.*

In other words, this corollary says that of all functions, EQ provides the largest separation between fooling set size and public-coin randomized communication complexity.

We next use the established connection between communication complexity of EQ and error correcting codes to provide a detailed analysis of the communication complexity k of EQ on n-bit inputs for varying amounts of random bits m and error ε. We start by translating bounds for error correcting codes to lower bounds for communication complexity. In particular, using the Singleton and Elias-Bassalygo bounds from coding theory, we obtain the following lower bound on the randomized communication complexity of EQ in the public-coin model (up to a constant factor):

Main Theorem 3 (Informal version of Theorem 8)**.** *Any randomized public-coin protocol for* EQ *satisfies the following lower bound for the communication complexity k:*

$$k \geq \max\left(\min\left(n, \frac{n}{\varepsilon 2^m}\right), \log\frac{1}{\varepsilon}\right).$$

Furthermore, this lower bound is tight for many regions of the parameters m, ε.

We show that the Plotkin bound and Griesmer bound do not give better bounds in any range of parameters (up to a constant factor).

Interestingly, the two known lower bounds for EQ mentioned above have natural interpretations in terms of codes. The result of [MSY16] corresponds to the Elias-Bassalygo bound (after translation of their result to the public randomness setting) and the result of [CG93] matches the Singleton bound. Our equivalence between communication protocols and error correcting codes provides a unified framework which captures both of these bounds. Furthermore, in our analysis of the equality function (Section 4), we notice that these bounds can be simplified (ignoring constant factors).

Next, we further analyze the landscape of communication complexity of EQ. Depending on values of the parameters m and ε, the expression in the theorem above can resolve (up to constant factors) to n, $\frac{n}{\varepsilon 2^m}$, or $\log\frac{1}{\varepsilon}$. In the first case, the trivial deterministic protocol for EQ (see Section 2.2) is clearly optimal. In the second case, we show that for a wide range of parameters, the EQ protocol obtained from Reed-Solomon codes is optimal. However, this code does not apply in a certain range of parameters, specifically when $\log\frac{1}{\varepsilon} < \frac{n}{\varepsilon 2^m} < m$. In this case, we devise a protocol that is suboptimal, but close to the lower bound (specifically, we iterate the Reed-Solomon code with itself). Finally, in the third case, we show that the standard protocol combined with an improvement of Newman's theorem (see Fact 5) for reducing the number of random bits gives an upper bound that is tight, up to additive lower order terms. Previously this was known only for the case of constant ε [CG93].

Other related work. Fleischer et al. [FJM95] studied the relation between the number of random bits and the zero-error communication complexity for the list-non-disjointness function. Ball et al. [BGM21] gave a lower bound on randomized communication complexity with defective randomness through the size of the fooling set. The paper [MSY16] analyses communication protocols of functions by looking at the statistical distance of the protocol on distinct inputs from the fooling set. The difference with our approach is that we look at Hamming distance between the vectors of outcomes instead.

The rest of the paper is organized as follows. In Section 2, we give the necessary definitions and state known results. In Section 3, we prove our results on the connection between communication complexity and error correcting codes. In Section 4, we use the connection established in Section 3 to deduce lower bounds on the communication complexity of equality for various amounts of allowed error and randomness. In Section 5, we analyze the landscape of communication complexity of equality by comparing our lower bounds against upper bounds for various regions of parameters.

2 Preliminaries

We make frequent use of the following fact:

Fact 1. *For two non-negative functions $a(n), b(n)$ such that $a(n) = b(n) + \log b(n)$,*

$$b(n) = a(n) - \log a(n) + O(1).$$

2.1 Notation and Conventions

All logarithms are base 2 unless otherwise specified. We use the notation $f^{(i)}$ to denote the function f iterated with itself i times.

For brevity, unless specified otherwise, when we refer to the "complexity" of a function, we are referring to the randomized communication complexity in the public-coin model. Similarly, when we refer to a "protocol", we mean a randomized communication protocol in the public-coin model. By "naive protocol" for EQ, we mean the deterministic protocol in which Alice sends her entire input, and by the "standard protocol" for EQ, we mean the standard randomized protocol used in the public-coin model, as described in Section 2.2. Finally, when we refer to a "code", we are referring to an error correcting code.

We use m to denote the amount (in bits) of public randomness in a communication protocol, and we use ε to denote the allowed error. When protocols have two-sided error, we use ε_1 to denote false positive error and ε_0 to denote false negative error. For convenience, we often use k to denote public-coin randomized communication complexity. That is, we let k denote CC^{pub}. We use the notation $[N, K, D]$ code to refer to a code with block length N, message length K, and distance D (see Section 2.3 for definitions of these parameters).

2.2 Communication Complexity

In the standard model of communication complexity, two players, Alice and Bob, wish to compute the value of a function $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ on some given input (x, y). The challenge is that Alice only knows $x \in \{0,1\}^n$ and Bob only knows $y \in \{0,1\}^n$, and thus the players need to communicate (i.e., send bits) to compute $f(x, y)$.

Definition 1 (protocol, value, cost, one-way)**.** A *communication protocol* Π is a binary tree where each internal node v is labeled either by a function $a_v : \{0,1\}^n \to \{0,1\}$ or by a function $b_v : \{0,1\}^n \to \{0,1\}$ (denoting Alice or Bob's message respectively) and each leaf node is labeled with an element in $\{0,1\}$.

The *value* of the protocol Π on input (x, y) is the label reached by traversing the tree according to the following rules: start at the root, and for each internal node v labeled by a_v (resp b_v), move left if $a_v(x) = 0$ (resp $b_v(y) = 0$) and right otherwise. The *cost* of Π on input (x, y) is the length of the path taken on input (x, y). The *cost* of Π is the height of the tree. We call a protocol *one-way* if the entire protocol consists of just one message.

Definition 2 (deterministic communication complexity)**.** The *deterministic communication complexity* $CC(f)$ of $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is the minimum cost of Π, over all protocols Π that compute f.

Fact 2 (deterministic communication complexity of EQ_n)**.** *For $x, y \in \{0,1\}^n$, the equality function, $\mathrm{EQ}_n(x, y)$, is defined to be 1 if $x = y$ and 0 otherwise. We have that $CC(\mathrm{EQ}_n) = n + 1$.*

The lower bound on $CC(\mathrm{EQ})$ can be proved using the fooling set method (see [KN97]). We define a fooling set below:

Definition 3 (fooling set [KN97])**.** A *fooling set* for $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is a set $F \subseteq \{0,1\}^n \times \{0,1\}^n$ and a value $b \in \{0,1\}$ such that

1. For every $(x, y) \in F$, $f(x, y) = b$
2. For all distinct pairs $(x_1, y_1), (x_2, y_2)$, either $f(x_1, y_2) \neq b$ or $f(x_2, y_1) \neq b$.

While Definition 1 defines protocols for deterministic communication complexity, this paper focuses on randomized communication complexity, wherein Alice and Bob have a source of public randomness. Formally, there is a random string of bits $r \in \{0,1\}^*$ available to both players. Where in the deterministic setting we had internal nodes in the protocol tree labeled with $a_v(x)$ and $b_v(y)$, we now have internal nodes labeled with $a_v(x, r)$ and $b_v(y, r)$; that is, Alice and Bob's communication can depend on r. We can also view this model as a distribution $\{P_r\}_{r\in\Pi}$ on deterministic protocols:

Definition 4 (randomized public-coin protocol [KN97])**.** A *(randomized) public-coin protocol* is a probability distribution over deterministic protocols. The *success probability* of a public-coin protocol on input (x, y) is the probability of choosing a (deterministic) protocol, according to the probability distribution $\{P_r\}_{r\in\Pi}$, that computes $f(x, y)$ correctly.

Definition 5 (randomized communication complexity [KN97])**.** The *randomized communication complexity* $CC^{\mathrm{pub}}_{\varepsilon}(f)$ is the minimum cost of a public-coin protocol that computes f with an error of at most ε (for every input (x, y)).

Fact 3 (randomized communication complexity of EQ)**.** *The randomized communication complexity (in the public randomness model) of* EQ *with constant error ε is $O(1)$.*

Fact 3 can be proven by considering the following protocol for EQ, which we call the "standard protocol" for EQ. Alice and Bob choose n bits of the public random string; call these bits r. Alice computes the inner product $b = \langle x, r\rangle$ over $\mathbb{F}_2$. She then sends this value (a single bit) to Bob. Bob checks whether $b = \langle y, r\rangle$ and sends 1 if they are equal and 0 otherwise. When $x = y$, Bob outputs

1 with probability 1. If $x \neq y$, Bob outputs 1 with probability $\frac{1}{2}$. By repeating the procedure t times with different n-bit random strings r, the error probability can be reduced to $\frac{1}{2^t}$.

Randomized protocols can be converted to deterministic protocols, with some overhead on the communication complexity, by repeating the randomized procedure and taking a majority vote. The following upper bound is proved in [CG93], and we include a proof here for completeness:

Fact 4 ([CG93])**.** *Let $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, be a function computable via a randomized communication protocol Π using k bits of communication, m bits of randomness, and with error probability ε. Then the deterministic communication complexity $CC(f)$ is bounded by*

$$CC(f) \leq k(2^m\varepsilon + 1). \tag{1}$$

Proof. Since is f computable by Π using m bits of randomness, k bits of communication, and error rate ε, for any input (x, y), there exist at most $\varepsilon 2^m$ choices of random bits such that the protocol fails to output $f(x, y)$. Thus, we can construct a deterministic protocol β by simulating Π over any $2\varepsilon 2^m + 1$ choices of predetermined unique random strings. The majority output will be the correct computation of f, and thus from the randomized protocol Π we have obtained a deterministic protocol for f using $k(2^m\varepsilon + 1)$ bits of communication. □

The following fact improves upon Newman's Theorem [New91], and states that a public-coin protocol can be converted into a private-coin protocol with only logarithmic additive overhead in the complexity. Similarly, the amount of randomness used by a public-coin protocol can be reduced by incurring a small amount of extra error.

Fact 5 ([LMdW21])**.** *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a function. Then $CC^{priv}_{\varepsilon(1+\delta)}(f) \leq CC^{pub}_{\varepsilon}(f) + \log(\frac{n}{\varepsilon}) + \log(\frac{6}{\delta^2})$. Furthermore, any public-coin protocol for f which uses m random bits and has error ϵ can be converted into a protocol using $\log n + \log(1/\varepsilon) + \log(\frac{6}{\delta^2})$ random bits with error $\epsilon(1 + \delta)$, at no penalty to the communication complexity.*

Interestingly, this improvement over original Newman's Theorem is crucial for tight bounds in one of the cases in our analysis in Section 5.3.

See [KN97] for more on private and public-coin settings in communication complexity.

2.3 Error Correcting Codes and Bounds

Definition 6 (code, block length, codewords)**.** A *code* C of *block length* N over alphabet Σ is $C \subseteq \Sigma^n$. The elements $c \in C$ are called *codewords*.

Definition 7 (distance, relative distance)**.** The *distance* D of a code $C \subseteq \Sigma^N$ is $\min_{c \neq c'}(\Delta(c, c'))$, where Δ is the Hamming distance function. In other words, the *distance* of a code is the minimum Hamming distance between any pair of codewords. The *relative distance* δ of C is the ratio of distance to block length, $\frac{D}{N}$.

Definition 8 (message length)**.** The *message length* (sometimes called *dimension*) K of a code C over alphabet Σ with $|\Sigma| = q$ is $K = \log_q |C|$.

We call a code with block length N, message length K, and distance D an $[N, K, D]$ code.

Definition 9 (rate)**.** The *rate* of a code with block length N and message length K is $R \coloneqq \frac{K}{N}$.

High distance and high rate are desirable properties for codes; however, there is a trade-off between these two parameters. The bounds stated below offer some insight into this trade-off.

Fact 6 (Singleton Bound)**.** *Let C be an $[N, K, D]$ code with alphabet size q. Then*

$$K \leq N - D + 1.$$

Equivalently, we can express the Singleton Bound in terms of rate and relative distance:

$$R \leq 1 - \delta + \frac{1}{N}.$$

The Singleton Bound is tight when $q \geq N$. That is, there exist codes for which $K = N - D + 1$ under this condition. We call such codes MDS (Maximum Distance Separable) codes. See Section 2.3 below.

Definition 10 (q-ary entropy)**.** The *q-ary entropy function* $H_q(\delta)$, a generalization of binary entropy, is defined as follows:

$$H_q(\delta) := \delta \log_q \left(\frac{q-1}{\delta}\right) + (1-\delta) \log_q \left(\frac{1}{1-\delta}\right).$$

As the block length N of a code tends to infinity, we have the relationship $\mathrm{Vol}_q(\delta N, N) \approx q^{NH_q(\delta)}$, and hence H_q appears in the asymptotic statement of various bounds for error-correcting codes, such as the Elias-Bassalygo Bound below:

Fact 7 (Elias-Bassalygo Bound)**.** *Let C be an $[N, K, D]$ code with alphabet size q. Then*

$$R := \frac{K}{N} \leq 1 - H_q\left[\left(1 - \frac{1}{q}\right)\left(1 - \sqrt{1 - (\frac{q}{q-1})\delta}\right)\right].$$

Fact 8 (Griesmer Bound)**.** *For a linear $[N, K, D]$ code with alphabet size q, we have:*

$$N \geq \sum_{i=0}^{K-1} \left\lceil \frac{D}{q^i} \right\rceil.$$

For more information on linear codes, see [Rot06].

Fact 9 (Plotkin Bound)**.** *The Plotkin bound strengthens the Singleton bound. For a code C with message length K, block length N, relative distance δ, and alphabet size q, we have:*

$$\frac{K}{N} + \delta\left(\frac{q}{q-1}\right) \leq 1 + \frac{1}{N}.$$

Our upper bounds use protocols based on Reed-Solomon codes, which we define below. For more information about these codes, see [Rot06].

Reed-Solomon codes are a family of codes which achieve the Singleton bound (i.e., for which $D = N - K + 1$).

Formally, Reed-Solomon codes are defined as follows:

Definition 11 (Reed-Solomon codes)**.** Let $q \geq N \geq K$. Let $\alpha_1, \ldots \alpha_N \in \mathbb{F}_q$ be distinct. The Reed-Solomon code over $\mathbb{F}_q$ with evaluation points $\overrightarrow{\alpha} = (\alpha_1, \ldots, \alpha_N)$, is the $[N, K, N-K+1]$ code given by:

$$\mathrm{RS}_q(N, K-1) = \{(f(\alpha_1), f(\alpha_2) \ldots, f(\alpha_N)) : f \in \mathbb{F}_q[X],\ \deg(f) \leq K-1\}.$$

2.4 Codes Give Communication Protocols for EQ

It is known (see [RY20, beginning of Chapter 3]) that codes give communication protocols for EQ. In particular, we have the following fact:

Fact 10. *For every $[N, K, D]$ code with alphabet Σ of size q, there exists a communication protocol, Π, computing EQ_n, such that $n \leq K \log q$, using m public random bits, and $\log q$ bits of communication, with error $\frac{N-D}{2^m}$. Moreover, this protocol is one-way and has one-sided false-positive error.*

Proof. Prior to the start of the communication protocol, Alice and Bob agree on a fixed subset $S \subseteq [N]$ such that $|S| = 2^m$. The protocol is roughly as follows. Alice and Bob translate their binary inputs to the Σ alphabet, and then pad appropriately to get $x', y' \in \Sigma^K$. They then encode their strings to get $f(x'), f(y') \in \Sigma^N$. Finally, using m bits of randomness, Alice and Bob select $i \in S$ and Alice shares $f(x')[i]$ with Bob. Bob outputs 1 if $f(x')[i] = f(y')[i]$, and otherwise outputs 0.

Clearly, if $x = y$, then $f(x')[i] = f(y')[i]$ for all indices i. If $x \neq y$ then the distinct codewords $f(x')$ and $f(y')$ will match on at most $N - D$ indices and thus will err at most on $\frac{N-D}{|S|} = \frac{N-D}{2^m}$ fraction of random choices. □

3 Communication Complexity vs. Error Correcting Codes

In this section, we show that randomized communication protocols for EQ give codes. In fact, we prove a generalization of this statement which applies to functions that are close to EQ in some sense, which we call "EQ-like" functions.

Definition 12 (EQ-like)**.** We say that a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is EQ-like if for all $x \in \{0,1\}^n$, $f(x,x) = 1$ and for all pairs (x,y) such that $x \neq y$, either $f(x,y) = 0$ or $f(y,x) = 0$.

Note that EQ is a particular example of an EQ-like function, where for all pairs (x,y) such that $x \neq y$, $f(x,y) = 0$ ***and*** $f(y,x) = 0$.

3.1 Protocols for EQ Give Codes

Consider a protocol for f. Alice holds x, Bob holds y, and there is public randomness r. Let $T(x,y,r)$ be the transcript of Alice and Bob's communication. That is,

$$T(x,y,r) = (a_1, b_1, a_2, b_2, \ldots, a_R, b_R),$$

where each a_i is a function of all previous messages, x, and their shared randomness r, and each b_i is a function of all previous messages, y, and r. Without loss of generality, at the end of the protocol, Alice outputs $c(T(x,y,r), x, r)$. Let $\varepsilon_0, \varepsilon_1$ denote the error parameters for the protocol; that is

$$\forall (x,y) \text{ such that } f(x,y) = 0, \quad \Pr_r[c(T(x,y,r),x,r) = 1] \leq \varepsilon_0$$
$$\forall (x,y) \text{ such that } f(x,y) = 1, \quad \Pr_r[c(T(x,y,r),x,r) = 1] \geq 1 - \varepsilon_1.$$

We show that the transcript T of Alice and Bob's messages forms a code. An encoding of $x \in \{0,1\}^n$ is simply the concatenation of $T(x,x,r)$ for all choices of r.

Lemma 1. *Let f be* EQ*-like, and let $T(x, y, r)$ be the transcript for Alice and Bob's communication in a randomized protocol for f as given above. Then for $x, y \in \{0,1\}^n$ with $x \neq y$,*

$$\Pr_r[T(x,x,r) = T(y,y,r)] \leq \varepsilon_0 + \varepsilon_1.$$

Proof. Let $x, y \in \{0,1\}^n$ with $x \neq y$. Without loss of generality suppose $f(x,y) = 0$ [1]. Then we have

$$\begin{aligned}
&\Pr_r[c(T(x,y,r),x,r) = 1] + \Pr_r[c(T(y,y,r),y,r) = 0] \\
\geq &\big(\Pr_r[c(T(x,y,r),x,r) = 1 \mid T(x,x,r) = T(y,y,r)] \\
&\quad + \Pr_r[c(T(y,y,r),y,r) = 0 \mid T(x,x,r) = T(y,y,r)]\big) \\
&\quad \cdot \Pr_r[T(x,x,r) = T(y,y,r)]. \\
= &\big(\Pr_r[c(T(y,y,r),y,r) = 1 \mid T(x,x,r) = T(y,y,r)] \\
&\quad + \Pr_r[c(T(y,y,r),y,r) = 0 \mid T(x,x,r) = T(y,y,r)]\big) \\
&\quad \cdot \Pr_r[T(x,x,r) = T(y,y,r)] \\
= &1 \cdot \Pr_r[T(x,x,r) = T(y,y,r)],
\end{aligned} \tag{2}$$

where the substitution $T(x,y,r) = T(y,y,r)$ follows from the fact that $T(x,x,r) = T(y,y,r)$ and that the communication tree partitions inputs into combinatorial rectangles.

Hence,

$$\Pr_r[T(x,x,r) = T(y,y,r)] \leq \Pr_r[c(T(x,y,r),x,r) = 1] + \Pr_r[c(T(y,y,r),y,r) = 0] \leq \varepsilon_0 + \varepsilon_1.$$

□

Using Lemma 1, we obtain the following theorem:

Theorem 2. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be* EQ*-like, and suppose there exists a randomized communication protocol for f which has communication complexity k, uses m bits of public randomness, and has two-sided error probabilities ε_0 and ε_1. Then there exists an error correcting code with message length n, codeword length 2^m, relative distance $1 - (\varepsilon_0 + \varepsilon_1)$, and alphabet size 2^k.*

Proof. The proof follows directly from Lemma 1. Concretely, the codeword for $x \in \{0,1\}^n$ is given by the concatenation of $T(x,x,r)$ for all 2^m possible values of randomness $r \in \{0,1\}^m$. Each symbol in the codeword corresponds to $T(x,x,r)$ for one choice of r, and since each transcript uses k bits of communication, there are 2^k possible symbols in the code alphabet. Finally, by Lemma 1, the probability that two different codewords agree on any given coordinate is bounded above by $\varepsilon_0 + \varepsilon_1$. Thus the relative distance of the code is $1 - (\varepsilon_0 + \varepsilon_1)$. □

If we consider a protocol with one-sided error, that is, a protocol with either $\varepsilon_0 = 0$ or $\varepsilon_1 = 0$, we immediately obtain the following corollary:

Corollary 3 (One-sided error)**.** *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be* EQ*-like, and suppose there exists a randomized communication protocol for f which has communication complexity k, uses m bits of public randomness, and has one-sided error probability ε. Then there exists a code with message length n, codeword length 2^m, relative distance $1 - \varepsilon$, and alphabet size 2^k.*

[1] If instead $f(y,x) = 0$, we replace $T(x,y,r)$ with $T(y,x,r)$ in the first two lines of Equation (2), and the proof holds.

We remark that since codes give protocols (Fact 10), this corollary shows that one-way protocols for EQ with one-sided false-positive error are in one-to-one correspondence with error correcting codes. That is, we can translate from protocols to codes and vice versa with no loss in parameters. Another consequence of Corollary 3 is the following corollary:

Corollary 4. *Any randomized protocol for* EQ *can be converted to a one-way protocol with one-sided false-positive error with no loss in parameters.*

The following table summarizes the relationship between the communication complexity of a protocol for EQ and the obtained code:

Communication Complexity	**Code**
size of input $= n$	# of messages $= 2^n$
# of random bits $= m$	length of code $= 2^m$
error $= \varepsilon$	relative distance $= 1 - \varepsilon$
complexity $= k$	alphabet size $= 2^k$

Figure 1: Translation Between CC and Codes

3.2 Codes from Functions Based on Fooling Sets

We have seen that protocols for EQ-like functions give codes. We can actually make a more general statement which applies to all functions. In particular, for any function f, a protocol for f gives a code whose parameters depend on the parameters of the protocol and the size of the largest fooling set for f.

Theorem 5. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be any function with fooling set F, and suppose there exists a randomized communication protocol for f which has communication complexity k, uses m bits of randomness, and has error probabilities $\varepsilon_0, \varepsilon_1$. Then there exists a code with block length 2^m, message length $\log |F|$, relative distance $1 - (\varepsilon_0 + \varepsilon_1)$, and alphabet size 2^k.*

Proof. To prove the theorem, we show how to construct an error correcting code from the function f with fooling set F. Let X and Y denote the set of x and y inputs respectively which appear in elements of the fooling set F, and define $f' : \{0,1\}^{\log |F|} \times \{0,1\}^{\log |F|} \to \{0,1\}$ to be f restricted to the inputs $X \times Y$. Without loss of generality, assume that for $(x, y) \in F$, $f(x, y) = 1$. By the definition of fooling set (see Section 2.2), f' is EQ-like, up to a permutation on one half of the input. Therefore, a protocol for computing f gives a protocol for f', which by Lemma 1 gives an error correcting code for messages of length $\log |F|$. □

As a consequence of Theorem 5, we obtain the following corollary, which says that in some sense, EQ represents the largest gap between fooling set size and randomized communication complexity in the public-coin model.

Corollary 6. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be any function with fooling set F, and suppose there exists a randomized communication protocol for f which has communication complexity k, uses m bits of randomness, and has error probabilities $\varepsilon_0, \varepsilon_1$. Then there exists a protocol for* EQ *on $\log |F|$ bits which has communication complexity k, uses m bits of randomness, and has one-sided false-positive error probability $\varepsilon_0 + \varepsilon_1$.*

This follows because, given a protocol for f, we can construct a code with message length $\log |F|$, from which we obtain a protocol for EQ on $\log |F|$ bits.

Informally, Corollary 6 says that among all functions with a fooling set of size s, EQ on $\log s$ bits has the smallest randomized complexity in the public-coin setting for all values of ε and m. For protocols with two-sided error, this interpretation is true up to constant factors. For protocols with one-sided error, it is precise.

4 Communication Lower Bounds Implied by Code Bounds

The equivalence between EQ communication protocol and codes allows us to view bounds on codes as bounds on communication complexity. In particular, since we showed that existence of protocols implies existence of codes, nonexistence of codes implies nonexistence of protocols. Using this connection, we translate known bounds for codes (Section 2.3) into bounds on communication complexity using the correspondence given by Figure 1[2].

4.1 Singleton Bound

Using the Singleton Bound for codes (Fact 6), we obtain the following bound for protocols:

$$2^m \geq \tfrac{n}{k} + 2^m(1-\varepsilon) - 1. \tag{3}$$

Rearranging yields

$$k \geq \frac{n}{1+\varepsilon 2^m}. \tag{4}$$

Depending on m and ε, either 1 or $\varepsilon 2^m$ is the dominant term in the denominator. Based on these two cases, the bound can be equivalently stated (up to constant factors) as

$$k \geq \min(\frac{n}{\varepsilon 2^m}, n). \tag{5}$$

Note that when n is the communication lower bound, it is optimal since it is achieved by the naive protocol for EQ.

This bound essentially coincides with the bound of [CG93] (see Fact 4).

4.2 Elias-Bassalygo Bound

Next, we apply the Elias-Bassalygo bound (Fact 7) to obtain the following bound for protocols:

$$\frac{n}{k \cdot 2^m} \leq 1 - H_{2^k}\left[\left(1-\frac{1}{2^k}\right)\cdot\left(1-\sqrt{1-\frac{2^k}{2^k-1}(1-\varepsilon)}\right)\right], \tag{6}$$

where H_{2^k} is the 2^k-ary entropy function as defined in Section 2.3. Now we use the fact that H_{2^k} has the following Taylor approximation [GRS23]:

$$H_{2^k}\left(1-\frac{1}{2^k}-\alpha\right) \geq 1 - c_{2^k}\alpha^2. \tag{7}$$

where the value $c_{2^k} = \frac{2^{2k}}{2(2^k-1)\ln(2^k)}$ depends only on k. Expanding yields

[2]Note that these lower bounds also hold for functions with large fooling sets.

$$\frac{n}{k \cdot 2^m} \leq 1 - H_{2^k}\left(1 - \frac{1}{2^k} - \underbrace{\left(\sqrt{1 - \frac{2^k}{2^k - 1}(1-\varepsilon)} - \frac{1}{2^k}\sqrt{1 - \frac{2^k}{2^k - 1}(1-\varepsilon)}\right)}_{\alpha}\right). \tag{8}$$

Substituting (8) in (6) and applying (7) gives

$$\begin{aligned}\frac{n}{k \cdot 2^m} &\leq 1 - c_{2k}\left(\sqrt{1 - \frac{2^k}{2^k - 1}(1-\varepsilon)} - \frac{1}{2^k}\sqrt{1 - \frac{2^k}{2^k - 1}(1-\varepsilon)}\right)^2 \\ &= \frac{2^{2k}}{2(2^k - 1)\ln(2^k)}\left(\frac{(2^k-1)^2}{2^{2k}}\right)\left(\frac{2^k\varepsilon - 1}{2^k - 1}\right) \\ &= \frac{2^k\varepsilon - 1}{2k\ln(2)}.\end{aligned} \tag{9}$$

Isolating the communication complexity k yields the following lower bound:

$$k \geq \log\left(\frac{2n\ln 2}{\varepsilon 2^m} + \frac{1}{\varepsilon}\right). \tag{10}$$

Up to constant additive terms, this is equivalent to

$$k \geq \max\left(\log\left(\frac{n\ln(2)}{\varepsilon 2^m}\right), \log\left(\frac{1}{\varepsilon}\right)\right), \tag{11}$$

which simplifies to

$$k \geq \max\left((\log(n) - m + \Theta(1)) + \left(\log\frac{1}{\varepsilon}\right), \log\frac{1}{\varepsilon}\right). \tag{12}$$

To further simplify this, we make the following observation: if the second term dominates in the expression $(\log(n) - m + \Theta(1)) + \left(\log\frac{1}{\varepsilon}\right)$, then, up to a constant factor, this bound is equivalent to $k \geq \log\frac{1}{\varepsilon}$. Since $\log\frac{1}{\varepsilon}$ already appears as a term in (12), we can further simplify the bound to

$$k \geq \max\left(\log(n) - m + \Theta(1), \log\frac{1}{\varepsilon}\right). \tag{13}$$

This bound matches (up to a constant factor) the public-coin translation of the bound given in [MSY16].

4.3 Griesmer Bound

The Griesmer Bound (Fact 8) offers a slight improvement to the Singleton bound and the Elias-Bassalygo bound for linear codes, but we show that this improvement is small. The Griesmer bound, translated using Figure 1, is given by

$$2^m \geq \sum_{i=0}^{\frac{n}{k}-1}\left\lceil\frac{2^m(1-\varepsilon)}{2^{ik}}\right\rceil. \tag{14}$$

We will show that even a slightly stronger lower bound is always within a constant factor of the Singleton or the Elias-Bassalygo bound. Note that the ceiling function can add at most 1 to each term in the sum. Thus the following bound is slightly stronger:

$$2^m \geq \frac{n}{k} + \sum_{i=0}^{\frac{n}{k}-1} \frac{2^m(1-\varepsilon)}{2^{ik}}. \tag{15}$$

Evaluating the geometric series yields

$$2^m \geq \frac{n}{k} + 2^m(1-\varepsilon)\left(\frac{1-2^{-n}}{1-2^{-k}}\right), \tag{16}$$

and rearranging allows us to lower bound the error rate by

$$\varepsilon \geq 2^{-k} + \frac{n2^{-m}}{k} - \frac{n}{k2^{m+k}}. \tag{17}$$

It suffices to consider only the positive terms in (17). In the event that $2^{-k} > \frac{n2^{-m}}{k}$, we obtain the Elias-Bassalygo bound up to a factor of 2:

$$\varepsilon > 2 \cdot \frac{1}{2^k}. \tag{18}$$

If $2^{-k} < \frac{n2^{-m}}{k}$, we recover the Singleton bound up to constant factors:

$$\varepsilon > 2 \cdot \frac{n}{k2^m}. \tag{19}$$

Therefore, up to constant factors in the error rate, the Griesmer bound does not allow us to achieve stronger lower bounds on the communication complexity.

4.4 Plotkin Bound

The Plotkin bound (Fact 9) can be expressed in terms of communication protocols as:

$$\frac{n}{k2^m} + (1-\varepsilon)\left(\frac{2^k}{2^k-1}\right) \leq 1. \tag{20}$$

In terms of error rate ε, this implies

$$\varepsilon \geq \underbrace{\frac{n}{2^m k}\left(\frac{2^k-1}{2^k}\right)}_{A} + \underbrace{\frac{1}{2^k}}_{B}. \tag{21}$$

However, this bound on the error rate is always within a constant factor of that given by the Singleton bound or Elias-Bassalygo bound. Indeed, if $A > B$, we have the following, which is strictly tighter than (21):

$$\varepsilon > 2 \cdot \frac{n}{2^m k}. \tag{22}$$

This bound recovers the error rate given by the Singleton bound (obtained by rearranging (4) to lower bound ε) up to constant factor. If $B > A$, the bound is given by

$$\varepsilon \geq \frac{2}{2^k}, \tag{23}$$

which recovers the error rate given by Elias-Bassalygo bound (13) up to a constant factor. This bound is also strictly tighter than (21). Therefore, up to constant factors in the error rate, the Plotkin bound does not allow us to achieve stronger lower bounds than we can achieve with the Singleton or Elias-Bassalygo bound.

4.5 Summarizing Lower Bounds

In this section, we present a unified analysis of our bounds. Directly combining the Singleton bound and Elias-Bassalygo bound results in the following:

$$k \geq \max\left(\min(n, \frac{n}{\varepsilon 2^m}), \log n - m + \Theta(1), \log\frac{1}{\varepsilon}\right), \tag{24}$$

which holds up to a constant factor.

However, the following lemma allows us to simplify the expression.

Lemma 7. *The maximum in (24) never evaluates to* $\log(n) - m + \Theta(1)$*. More precisely, if the* $\log(n) - m + \Theta(1)$ *term is maximal, some other term is also maximal (up to a constant factor).*

Proof. Indeed, if the $\log(n) - m + \Theta(1)$ term is maximal, we have the following system:

$$\begin{cases} \log(n) - m + \Theta(1) \geq \log\frac{1}{\varepsilon}, \\ \log(n) - m + \Theta(1) \geq \min(n, \frac{n}{\varepsilon 2^m}). \end{cases}$$

The inequality $\log(n) - m + \Theta(1) \geq n$ is impossible, which leaves us with

$$\begin{cases} \log(n) - m + \Theta(1) \geq \log\frac{1}{\varepsilon}, \\ \log(n) - m + \Theta(1) \geq \frac{n}{\varepsilon 2^m}. \end{cases}$$

Let $A = \log(n) - m$ and $B = \log\frac{1}{\varepsilon}$. Then, taking the logarithm of the second inequality, we get

$$\begin{cases} A + \Theta(1) \geq B, \\ \log(A + \Theta(1)) \geq B + A. \end{cases}$$

Note that we can assume that $\varepsilon \leq \frac{1}{2}$ and thus $B > 0$. Now we apply this to the second inequality to obtain

$$\log(A + \Theta(1)) \geq A,$$

which implies that A is constant. Then, it follows from the first inequality that B is also constant. Overall, we have that third term in (24) is also constant. □

Thus, we can simplify (24) to obtain the following theorem.

Theorem 8. *For any randomized public-coin protocol for* EQ *on* n *bits with complexity* k*,* m *bits of randomness, and error probability* ε*,*

$$k \geq \max\left(\min(n, \frac{n}{\varepsilon 2^m}), \log\frac{1}{\varepsilon}\right). \tag{25}$$

Note that Theorem 8 gives a different lower bound (one of n, $\frac{n}{\varepsilon 2^m}$, or $\log \frac{1}{\varepsilon}$) for different regions of the parameters n, m, and ε. In Section 5, we examine the necessary relationship between n, m, and ε for achieving each of the three possible lower bounds. We also analyze when these bounds are tight by giving matching upper bounds for most regions of the parameters. This analysis is summarized in the table below.

Description of Region	**Lower Bound**	**Upper Bound**
$\frac{1}{2^m} \geq \varepsilon \geq \frac{1}{2^n}$	n	n
$n \geq \frac{n}{\varepsilon 2^m} \geq \log \frac{1}{\varepsilon}$	$\frac{n}{\varepsilon 2^m}$	$\frac{n}{\varepsilon 2^m}$, if $m < \frac{n}{\varepsilon 2^m}$
		$\frac{n}{2^m}$, if $\log n - \log^{(c)} n > m > \frac{n}{\varepsilon 2^m}$ and constant ε and c
		$\frac{n}{\varepsilon^c 2^m}$, if $\varepsilon = \frac{1}{\log^{(c)} n}$ and c is constant
$\log n + \log \frac{1}{\varepsilon} - \log \log \frac{1}{\varepsilon} < m$	$\log \frac{1}{\varepsilon}$	$\log \frac{1}{\varepsilon}$, if $m > \log n + \log \frac{1}{\varepsilon} + \Theta(1)$

Figure 2: Bounds on complexity of EQ for various regions of parameters

5 Analyzing the Landscape of Communication Complexity of EQ

In this section, we analyze the various regions of parameters which give rise to different lower bounds on communication complexity. We further examine when these bounds are tight by giving upper bounds. Now we consider the three cases based on which of the three terms the maximum in Theorem 8 evaluates to.

5.1 Case 1

In this case, we have

$$\begin{cases} n \leq \frac{n}{\varepsilon 2^m}, \\ n \geq \log \frac{1}{\varepsilon}, \end{cases}$$

or, after rearranging, $\frac{1}{2^n} \leq \varepsilon \leq \frac{1}{2^m}$. In this case, $\max\left(\min(n, \frac{n}{\varepsilon 2^m}), \log \frac{1}{\varepsilon}\right) = n$, giving the lower bound $k \geq n$. Clearly, the naive protocol for EQ gives us the matching upper bound.

Remark 1. *This lower bound is actually easy to see, since for $\varepsilon < \frac{1}{2^m}$ only the deterministic protocol is possible (because of the granularity of error probability).*

5.2 Case 2

In this case, we have

$$\begin{cases} n \geq \frac{n}{\varepsilon 2^m}, \\ \frac{n}{\varepsilon 2^m} \geq \log \frac{1}{\varepsilon}. \end{cases}$$

The lower bound in this case is $k \geq \frac{n}{\varepsilon 2^m}$.

Remark 2. *Note that this case shows that we cannot study the value of m up to a constant factor. The reason is that the bound depends on 2^m and multiplicative changes for m may affect the bound dramatically. Note that we never allowed extra multiplicative constant factors for m.*

5.2.1 Upper Bound with Reed-Solomon Protocol

For the upper bound in this case, we can use the communication protocol derived from Reed-Solomon codes (Section 2.3). In this protocol, Alice and Bob encode their inputs using a Reed-Solomon code and check the value of a single random index in a subset S of their respective codewords. For some values of parameters, the communication complexity of this protocol matches the lower bound given by the Singleton bound.

Lemma 9. *For any $m \leq k$, there exists a public-coin randomized communication protocol that computes* EQ *using k bits of communication, m bits of randomness, and error rate $\varepsilon = O(\frac{n}{k2^m})$.*

Proof. Let q be the largest prime number such that $q < 2^k$. To compute EQ, we use a $\mathrm{RS}_q(q, \lceil \frac{n}{k} \rceil)$ code to get an error rate of

$$\varepsilon = \frac{\lceil \frac{n}{k} \rceil}{2^m} = \frac{n}{k2^m} + \Theta\left(\frac{1}{2^m}\right), \tag{26}$$

but we know that $k \leq n + 1$ (the naive protocol for EQ gives this upper bound), giving $\varepsilon = O(\frac{n}{k2^m})$. □

Note that Reed-Solomon codes are only defined for $m \leq k$. For this protocol, $k = \Theta(\frac{n}{\varepsilon 2^m})$. Thus, we have a protocol only in the region

$$m \leq \Theta(\frac{n}{\varepsilon 2^m}).$$

Hence, this protocol does not give an upper bound when

$$\log \frac{1}{\varepsilon} \leq \frac{n}{\varepsilon 2^m} \leq m. \tag{27}$$

An example parameters in this region is $m = \log n$ and $\varepsilon = 1/\log\log n$. The complexity in this case is $\frac{n}{\varepsilon 2^m} = \log\log n$. We partially address this issue in the next section.

5.2.2 Upper bound with iterated Reed-Solomon

The Reed-Solomon code allows us to do the following transformation of parameters:

$$n \longmapsto (m, \varepsilon, \frac{n}{\varepsilon 2^m}).$$

This notation means that when we need to send a message of length n, we can use m random bits, introduce an error ε, and reduce the problem to sending a message of size $\frac{n}{\varepsilon 2^m}$. This protocol gives a tight upper bound, but it has the restriction $m \leq \frac{n}{\varepsilon 2^m}$. For the range of parameters

$$\log \frac{1}{\varepsilon} < \frac{n}{\varepsilon 2^m} < m,$$

we do not have a tight upper bound.

Restating in terms of m, Reed-Solomon codes give tight upper bounds for the region of parameters satisfying

$$m \leq \log n + \log \frac{1}{\varepsilon} - \log\left(\log n + \log \frac{1}{\varepsilon}\right). \tag{28}$$

Using Fact 1, we see that the following range of the lower bound is not matched with a tight upper bound:

$$\log n + \log \frac{1}{\varepsilon} - \log\left(\log n + \log \frac{1}{\varepsilon}\right) < m < \log n + \log \frac{1}{\varepsilon} - \log\log \frac{1}{\varepsilon}. \tag{29}$$

Our plan is to apply Reed-Solomon code iteratively. That is, in the first step, we apply the code with some parameters:

$$n \longmapsto (m_1, \varepsilon_1, \frac{n}{\varepsilon_1 2^{m_1}}),$$

but then instead of sending $\frac{n}{\varepsilon_1 2^{m_1}}$ bits, we apply Reed-Solomon code (with some new values of parameters) to this message again:

$$n_2 = \frac{n}{\varepsilon_1 2^{m_1}} \longmapsto (m_2, \varepsilon_2, \frac{n_2}{\varepsilon_2 2^{m_2}}).$$

Below we give the parameters of the resulting procedure. Notice that in both iterations we use random bits, but only communicate according to the result of the final iteration. The resulting procedure for two iterations gives us the following parameters

$$n \longmapsto \left(m = m_1 + m_2, \varepsilon \leq \varepsilon_1 + \varepsilon_2, \frac{n}{\varepsilon_1\varepsilon_2 2^{m_1+m_2}}\right),$$

and has the following restrictions

$$\begin{aligned} m_1 &\leq \frac{n}{\varepsilon_1 2^{m_1}}, \\ m_2 &\leq \frac{n}{\varepsilon_1\varepsilon_2 2^{m_1+m_2}}. \end{aligned}$$

Observe that we are interested in ε and communication complexity only up to a constant factor. Further, notice that our errors are only additive from the first to the second iteration. Finally, we see that allowing a higher error rate in a given round will never increase the amount of randomness or communication required. Therefore, we can assume that in each iteration we allow the same error ε; otherwise, we could simply choose to increase the allowed error in the iteration with a smaller error rate.

With this assumption, on an arbitrary iteration i we will have parameters

$$(m_i, \varepsilon, \frac{n}{\varepsilon^i 2^{m_1+\ldots+m_i}})$$

with the restriction

$$m_i \leq \frac{n}{\varepsilon^i 2^{m_1+\ldots+m_i}}. \tag{30}$$

The total values of parameters after c iterations (we assume that c is a constant) are

$$n \longmapsto \left(m = m_1 + \ldots + m_c, \varepsilon, \frac{n}{\varepsilon^c 2^{m_1+\ldots+m_c}}\right).$$

Recall that our protocol using single-iteration Reed-Solomon works only for $m < \frac{n}{\varepsilon 2^m}$. Thus our goal is to develop a protocol that is valid for larger values of m. Notice that only the final restriction directly contains a restriction on the total number of random bits used in the protocol:

$$m_c \le \frac{n}{\varepsilon^c 2^{m_1+\cdots+m_c}} = \frac{n}{\varepsilon^c 2^m}. \tag{31}$$

Rearranging this expression, we get

$$m \le \frac{n}{\varepsilon^c 2^{m_c}}.$$

We can see that minimizing the randomness used in the last iteration, m_c, while keeping m the same maximizes the range in which this protocol is valid. Naturally to minimize m_c, we need to maximize the randomness used in all previous iterations. Thus, to optimize the parameters we need to fix

$$m_i = \frac{n}{\varepsilon^i 2^{m_1+\ldots+m_i}}$$

on intermediate iterations.

For a real nonnegative x, we introduce the function $f(x) = \log(\log \frac{1}{\varepsilon} + x)$. We now describe the values of parameters that we can cover with c iterations of Reed-Solomon protocol.

Lemma 10. *For any integer constant c we have that the communication protocol derived from c iterations of Reed-Solomon has complexity $\frac{n}{\varepsilon^c 2^m}$ and is valid for all*

$$m \le \log n + c \log \frac{1}{\varepsilon} - f^{(c)}(\log n) \tag{32}$$

(up to an additive constant).

Moreover, if all restrictions (30) are saturated, the inequality (32) is saturated as well.

Proof. We prove this by induction on c. The base case corresponds to the standard Reed-Solomon protocol, which we already covered.

For the induction step, we do one extra step of the iteration and obtain the restriction

$$m_{c+1} \le \frac{n}{\varepsilon^{c+1} 2^{m_1+\ldots+m_{c+1}}}.$$

Since for the previous iterations we need to maximize $m_1, \ldots, m_c$, we must fix them so that all restrictions (30) are saturated. By the inductive hypothesis we have that (32) is saturated. Substituting $m_1 + \ldots + m_c$ from this equality to above yields

$$m_{c+1} \le \frac{2^{f^{(c)}(\log n)}}{\varepsilon 2^{m_{c+1}}}.$$

Rearranging this inequality we obtain, using Fact 1,

$$m_{c+1} \le \log \frac{1}{\varepsilon} + f^{(c)}(\log n) - \log\left(\frac{1}{\varepsilon} + f^{(c)}(\log n)\right) = \log \frac{1}{\varepsilon} + f^{(c)}(\log n) - f^{(c+1)}(\log n).$$

Adding this to the expression for $m_1 + \ldots + m_c$ gives the desired inequality.

Moreover, it is easy to see that if (30) is saturated for $i = c + 1$ as well, the resulting inequlity is saturated. □

As a result we get a protocol with complexity $\frac{n}{\varepsilon^c 2^m}$ for constant c and for parameters satisfying (32). For example, for constant ε the lower bound we get is still tight up to the constant factor. It is not hard to see in this case that the function $f^{(c)}(\log n)$ is equal to $\log^{(c)} n$ up to an additive constant, and thus the range of parameters for which we do not have an upper bound reduces to

$$\log n - \log^{(c)} n < m < \log n.$$

For non-constant ε, our upper bound is not tight. To analyze the interval in which it applies, observe that $f(x) = \log(\log \frac{1}{\varepsilon} + x)$ is equal to $\log \log \frac{1}{\varepsilon}$ (up to an additive constant 1) for $\log \frac{1}{\varepsilon} \geq x$ and is equal to $\log x$ (up to an additive constant 1) for $\log \frac{1}{\varepsilon} \leq x$.

For example, for $\varepsilon = \frac{1}{\log^{(t)} n}$, for constant t, we get that $\log \frac{1}{\varepsilon} = \log^{(t+1)} n$. After $t+1$ iterations we have that $f^{(t+1)} = \log \log \frac{1}{\varepsilon}$ (up to an additive constant 1). Substituting this into (32) we get that this upper bound covers the whole region (29).

5.3 Case 3

This case applies when

$$\log \frac{1}{\varepsilon} \geq n$$

or

$$\log \frac{1}{\varepsilon} \geq \frac{n}{\varepsilon 2^m}.$$

The lower bound in this case is $k \geq \log \frac{1}{\varepsilon}$.

Note that the first of the subcases here corresponds to $\varepsilon \leq \frac{1}{2^n}$, in which case the communication complexity bound is n. We know that this bound is tight.

The remaining case to consider is

$$\log \frac{1}{\varepsilon} \geq \frac{n}{\varepsilon 2^m}.$$

which, after the rearranging gives

$$2^m \geq n \frac{1}{\varepsilon \log \frac{1}{\varepsilon}}$$

or equivalently,

$$m \geq \log n + \log \frac{1}{\varepsilon} - \log \log \frac{1}{\varepsilon}.$$

As for the upper bound, we use the standard randomized protocol and apply Fact 5. The standard protocol gives us the parameters

$$(n \log \frac{1}{\varepsilon}, \varepsilon, \log \frac{1}{\varepsilon}).$$

Applying Fact 5 with error parameter δ gives

$$(\log n + \log \frac{1}{\varepsilon} + \log \left(\frac{6}{\delta^2} \right), \varepsilon(1+\delta), \log \frac{1}{\varepsilon}).$$

Setting $\delta = \Theta(1)$ yields

$$(\log n + \log \frac{1}{\varepsilon} + \Theta(1), \Theta(\varepsilon), \log \frac{1}{\varepsilon}),$$

and this bound achieves optimal m up to an additive factor in lower order terms, and optimal ε up to a constant multiplicative factor.

Remark 3. *We observe that this public-coin protocol implies a private-coin protocol, where the randomness is simply shared at the start of the protocol at the cost of* $\log n + \log \frac{1}{\varepsilon} + \Theta(1)$ *bits of communication. This resulting private-coin protocol has* $\Theta(\varepsilon)$ *error using* $\log(\frac{n}{\varepsilon^2}) + \Theta(1)$ *bits of communication, essentially matching the lower bound of* $\log(\frac{n}{\varepsilon^2}) - \log\log(\frac{1}{\varepsilon})$ *observed in [LMdW21]. This indicates that in this range of parameters, the optimal private-coin protocol is to simply share the private randomness and then simulate the standard public-coin protocol.*

6 Discussion & Future Directions

We identify an equivalence between randomized public-coin protocols for equality and error correcting codes, and generalize one direction of this equivalence to functions with large fooling sets. We use this connection to study the randomized communication complexity of EQ for various regions of parameters error ε and randomness m. We give lower bounds for protocols from lower bounds from codes, and analyze when these lower bounds are tight. An immediate question is whether we can tighten our bounds in the region in which they are not tight.

One way to view our result on fooling sets is that the maximal separation between the fooling set size and randomized communication complexity with public randomness is achieved by the equality function (for all regions of parameters). A natural future direction is to explore whether similar separations can be found between randomized communication complexity and other complexity measures, such as deterministic communication complexity, rectangle size, or rank. Partial progress in this direction was obtained by Canetti and Goldreich [CG93].

References

[BGM21] Marshall Ball, Oded Goldreich, and Tal Malkin. Communication complexity with defective randomness. In Valentine Kabanets, editor, *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPIcs*, pages 14:1–14:10. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[CG93] Ran Canetti and Oded Goldreich. Bounds on tradeoffs between randomness and communication complexity. *Comput. Complex.*, 3:141–167, 1993.

[CLV19] Arkadev Chattopadhyay, Shachar Lovett, and Marc Vinyals. Equality alone does not simulate randomness. In Amir Shpilka, editor, *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, volume 137 of *LIPIcs*, pages 14:1–14:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[dRMN+20] Susanna F. de Rezende, Or Meir, Jakob Nordström, Toniann Pitassi, Robert Robere, and Marc Vinyals. Lifting with simple gadgets and applications to circuit and proof complexity. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 24–30. IEEE, 2020.

[FJM95] Rudolf Fleischer, Hermann Jung, and Kurt Mehlhorn. A communication-randomness tradeoff for two-processor systems. *Inf. Comput.*, 116(2):155–161, 1995.

[FKNN95] Tomás Feder, Eyal Kushilevitz, Moni Naor, and Noam Nisan. Amortized communication complexity. *SIAM J. Comput.*, 24(4):736–750, 1995.

[GRS23] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. *Essential Coding Theory*. 2023.

[Juk12] Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*, volume 27 of *Algorithms and combinatorics*. Springer, 2012.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

[LM19] Bruno Loff and Sagnik Mukhopadhyay. Lifting theorems for equality. In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, volume 126 of *LIPIcs*, pages 50:1–50:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[LMdW21] Olivier Lalonde, Nikhil S Mande, and Ronald de Wolf. Tight bounds for the randomized and quantum communication complexities of equality with small error. *arXiv preprint arXiv:2107.11806*, 2021.

[MSY16] Shay Moran, Makrand Sinha, and Amir Yehudayoff. Fooling pairs in randomized communication complexity. In Jukka Suomela, editor, *Structural Information and Communication Complexity - 23rd International Colloquium, SIROCCO 2016, Helsinki, Finland, July 19-21, 2016, Revised Selected Papers*, volume 9988 of *Lecture Notes in Computer Science*, pages 49–59, 2016.

[New91] Ilan Newman. Private vs. common random bits in communication complexity. *Information Processing Letters*, 39(2):67–71, 1991.

[Nis93] Noam Nisan. The communication complexity of threshold gates. *Combinatorics, Paul Erdős Is Eighty*, page 301–315, 1993.

[PSS23] Toniann Pitassi, Morgan Shirley, and Adi Shraibman. The strength of equality oracles in communication. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 89:1–89:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[Raz11] Alexander A. Razborov. *Communication Complexity*, pages 97–117. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[Rot06] Ron M. Roth. *Introduction to coding theory*. Cambridge University Press, 2006.

[Rou16] Tim Roughgarden. Communication complexity (for algorithm designers). *Found. Trends Theor. Comput. Sci.*, 11(3-4):217–404, 2016.

[RY20] Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020.

[Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In Michael J. Fischer, Richard A. DeMillo, Nancy A. Lynch, Walter A. Burkhard, and Alfred V. Aho, editors, *Proceedings of the 11h Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA*, pages 209–213. ACM, 1979.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il