Better Boosting of Communication Oracles, or Not

Nathaniel Harms
EPFL, Lausanne, Switzerland

Artur Riazanov
EPFL, Lausanne, Switzerland

Abstract

Suppose we have a two-party communication protocol for f which allows the parties to make queries to an oracle computing g; for example, they may query an Equality oracle. To translate this protocol into a randomized protocol, we must replace the oracle with a randomized subroutine for solving g. If q queries are made, the standard technique requires that we boost the error of each subroutine down to $O(1/q)$, leading to communication complexity which grows as $q \log q$. For which oracles g can this naïve boosting technique be improved?

We focus on the oracles which can be computed by constant-cost randomized protocols, and show that the naïve boosting strategy can be improved for the Equality oracle but not the 1-Hamming Distance oracle. Two surprising consequences are (1) a new example of a problem where the cost of computing k independent copies grows superlinear in k, drastically simplifying the only previous example due to Blais & Brody (CCC 2019); and (2) a new proof that Equality is not complete for the class of constant-cost randomized communication (Harms, Wild, & Zamaraev, STOC 2022; Hambardzumyan, Hatami, & Hatami, Israel Journal of Mathematics 2022).

2012 ACM Subject Classification Theory of computation → Communication complexity

Keywords and phrases oracles, error reduction, communication complexity

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2024.25

Funding *Nathaniel Harms*: Supported by an NSERC postdoctoral fellowship and the Swiss State Secretariat for Education, Research, and Innovation (SERI) under contract number MB22.00026.
Artur Riazanov: Supported by the Swiss State Secretariat for Education, Research, and Innovation (SERI) under contract number MB22.00026.

1 Introduction

We typically require that randomized algorithms succeed with probability 2/3, since the probability can be boosted to any $1 - \delta$ by taking a majority vote of $O(\log(1/\delta))$ repetitions. If many randomized subroutines are used within an algorithm, the probability of error may accumulate, and one may apply standard boosting to each subroutine to bring the error probability down to an acceptable level. We wish to understand when this is necessary, in the setting of communication complexity.

Suppose two parties, Alice and Bob, wish to compute a function $f(x, y)$ on their respective inputs x and y, using as little communication as possible, and they have access to a shared (i.e. public) source of randomness. A convenient way to design a randomized communication protocol to compute $f(x, y)$ is to design a *deterministic* protocol, but assume that Alice and Bob have access to an *oracle* (in other words, a subroutine) which computes a certain problem g that itself has an efficient randomized protocol.

▶ **Example 1.** The Equality problem is the textbook example of a problem with an efficient randomized protocol [19, 23]: Given inputs $a, b \in [N]$, two parties can decide (with success probability 3/4) whether $a = b$, using only 2 bits of (public) randomized communication, regardless of the domain size N. So, to design a randomized protocol for solving another problem $f(x, y)$, we may assume that the two parties have access to an Equality oracle.

44th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2024).
Editors: Siddharth Barman and Sławomir Lasota; Article No. 25; pp. 25:1–25:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

For example, suppose Alice and Bob have vertices x and y in a shared tree T, and wish to decide whether x and y are adjacent in T. If $p(x)$ denotes the parent of x in T, then Alice and Bob can decide adjacency using two EQUALITY queries: "$x = p(y)$?" and "$y = p(x)$?"

▶ **Example 2.** The 1-HAMMING DISTANCE communication problem is denoted HD_1 and defined as $\mathrm{HD}_1^n(x, y) = 1$ if $x, y \in \{0,1\}^n$ differ on exactly 1 bit, and 0 otherwise. It has a constant-cost randomized protocol, but unlike adjacency in trees, this protocol *cannot* be expressed as a deterministic protocol using the EQUALITY oracle [12, 13].

Using oracles makes the protocol simpler, and also makes it clearer how and why randomness is used in the protocol, which provides more insight into randomized communication (see e.g. [5, 12, 13, 8] for recent work using oracles to understand randomized communication). But when we replace the *oracle* for g with a randomized protocol for g, we must compensate for the probability that the randomized protocol produces an incorrect answer. Write $\mathsf{D}^g(f)$ for the optimal cost of a deterministic communication protocol for f using an oracle for g (where the players pay cost 1 to query the oracle). Write $\mathsf{R}_\delta(f)$ for the optimal cost of a randomized protocol for f with error δ. Then the inequality

$$\forall f\ , \qquad \mathsf{R}_\delta(f) = O\left(\mathsf{D}^g(f) \cdot \mathsf{R}_{1/4}(g) \cdot \log\left(\frac{\mathsf{D}^g(f)}{\delta}\right)\right) \tag{1}$$

follows from standard boosting: if there are $q = \mathsf{D}^g(f)$ queries made by the protocol in the worst case, then we obtain a randomized protocol by simulating each of the q queries to g using a protocol for g with error $\approx \delta/q$, sending $\mathsf{R}_{\delta/q}(g) = O(\mathsf{R}_{1/4}(g) \cdot \log(q/\delta))$ bits of communication for each query. But is it possible to improve on this naïve bound? The main question of this paper is:

▶ **Question 3.** *For which oracle functions g can Equation* (1) *be improved?*

We focus on the oracles g which have *constant-cost* randomized communication protocols, like EQUALITY. Randomized communication is quite poorly understood, with many fundamental questions remaining open even when restricted to the surprisingly rich class of constant-cost problems. Many recent works have focused on understanding these extreme examples of efficient randomized computation; see [12, 13, 11, 6, 16, 14, 8] and the survey [15]. And indeed some of these works [12, 15] use Equation (1) specifically for the EQUALITY oracle. So this is a good place to begin studying Question 3. Our main result is:

▶ **Theorem 4** (Informal; see Theorems 11 and 14)**.** *Equation* (1) *can be improved for the* EQUALITY *oracle, but it is (nearly) tight for the 1-*HAMMING DISTANCE *oracle.*

This has some unexpected consequences, described below, and also answers Question 3 for all *known* constant-cost problems.

Every known constant-cost problem g satisfies either $\mathsf{D}^{\mathrm{EQ}}(g) = O(1)$ or $\mathsf{D}^g(\mathrm{HD}_1) = O(1)$ ([7] gives a survey of all known problems). Therefore we answer Question 3 for all *known* constant-cost oracles. Towards an answer for *all* constant-cost oracles, we show that the technique which allows us to improve Equation (1) works *only* for the EQUALITY oracle (Proposition 21).

Our main proof also has two other surprising consequences:

Direct sums. Direct sum questions ask how the complexity of computing k copies of a problem grows with k (see e.g. [9, 18, 1, 3]). Recently, [3] answered a long-standing question of [9] by providing the first example of a problem where the communication complexity of computing k independent copies grows *superlinearly* with k. Their example is specially designed to exhibit this behaviour and goes through the query-to-communication lifting technique. In our investigation of Question 3, we show that computing k independent copies of the drastically simpler, constant-cost 1-Hamming Distance problem requires $\Omega\left(\frac{k \log k}{\log \log k}\right)$ bits of communication (Theorem 14). As a corollary, we also show a similar direct sum theorem for randomized parity decision trees (Corollary 15).

Oracle separations. In an effort to better understand the power of randomness in communication, recent works have studied the relative power of different oracles. [5] show that the Equality oracle is not powerful enough to simulate the standard communication complexity class BPP (i.e. $N \times N$ communication matrices with cost poly $\log \log N$), i.e. Equality is not *complete* for BPP. [12, 13] showed that Equality is also not complete for the class BPP^0 of *constant-cost* communication problems, because 1-Hamming Distance does not reduce to it; and [8] show that there is *no* complete problem for BPP^0. There are many lower-bound techniques for communication complexity, but not many lower bounds for communication with *oracles*. Our investigation of Question 3 gives an unexpected new proof of the separation between the Equality and 1-Hamming Distance oracles; our proof is "algorithmic", and arguably simpler than the Ramsey-theoretic proof of [13] or the Fourier-analytic proof of [12].

Further Motivation, Discussion & Open Problems

Let's say a constant-cost oracle function g has *better boosting* if

$$\forall f \ : \qquad \mathsf{R}_\delta(f) = O(\mathsf{D}^g(f) + \log(1/\delta))\,.$$

We showed that among the *currently-known* constant-cost oracle functions g, better boosting is possible if and only if $\mathsf{D}^{\mathrm{EQ}}(g) = O(1)$, and we observed that among *all* constant-cost oracles, only the Equality oracle satisfies the properties used to prove Theorem 11. So, permit us the following conjecture:

▶ **Conjecture 5.** *An oracle function $g \in \mathsf{BPP}^0$ has better boosting if and only if $\mathsf{D}^{\mathrm{EQ}}(g) = O(1)$.*

To disprove this conjecture, we need a new example of a constant-cost (total) communication problem that is not somehow a generalization of 1-Hamming Distance. Such an example would be very interesting, so in that regard we hope the conjecture is false.

One more motivation of the current study is to find an approach towards a question of [14] about the intersection between communication complexity classes $\mathsf{UPP}^0 \cap \mathsf{BPP}^0$, where UPP^0 denotes the class of problems with bounded sign-rank, or equivalently, constant-cost *unbounded-error* randomized protocols [22]. Writing EQ^0 for the class of problems g where $\mathsf{D}^{\mathrm{EQ}}(g) = O(1)$, [14] asks:

▶ **Question 6** ([14]). *Is $\mathsf{UPP}^0 \cap \mathsf{BPP}^0 = \mathsf{EQ}^0$?*

This question seems challenging; as noted in [14], a positive answer would imply other conjectures about $\mathsf{UPP}^0 \cap \mathsf{BPP}^0$, notably the conjecture of [16] that 1-Hamming Distance does not belong to UPP^0, which would be the first example of a problem in $\mathsf{BPP}^0 \setminus \mathsf{UPP}^0$.

[16] showed that all known lower-bound techniques against UPP^0 fail to prove this. But a positive answer to Question 6 implies that all oracles in $\mathsf{UPP}^0 \cap \mathsf{BPP}^0$ have better boosting, so a weaker question is:

▶ **Question 7.** *Do all oracles in* $\mathsf{UPP}^0 \cap \mathsf{BPP}^0$ *have better boosting?*

Because of Theorem 4, this weaker question would also suffice to prove that 1-Hamming Distance does not belong to UPP^0. It is not clear to us whether Question 7 is easier to answer than Question 6. If the answer to Question 7 is negative (i.e. there is an oracle in $\mathsf{UPP}^0 \cap \mathsf{BPP}^0$ which does not have better boosting), then either Conjecture 5 or Question 6 is false.

Similar questions about probability boosting were studied recently for query complexity in [2] who focused on the properties of the *outer* function f of which allow for better boosting to compute $f \circ g^{\otimes k}$, whereas one may think of our oracles as the *inner* functions. We may rephrase Theorem 11 as a "composition theorem" which says that for any function $f\colon \{0,1\}^k \to \{0,1\}$, the composed function $f \circ (\text{Eq})^{\otimes k}$ which applies f to the result of k instances of Equality has communication cost

$$\mathsf{R}_\delta(f \circ (\text{Eq})^{\otimes k}) = O(\mathsf{DT}(f) + \log(1/\delta)) \tag{2}$$

where $\mathsf{DT}(f)$ is the decision-tree depth of f. We prefer the statement in Theorem 11 because it more clearly differentiates between the *protocol* and the *problem*. To see what we mean, consider taking f to be the And function; the immediate consequence of Equation (2) is that $\mathsf{R}_{1/4}(\text{And} \circ (\text{Eq})^{\otimes k}) = O(k)$, whereas the immediate consequence of Theorem 11 is that $\mathsf{R}_{1/4}(\text{And} \circ (\text{Eq})^{\otimes k}) = O(1)$ because this function can be computed using 1 Equality query. To get the same result from Equation (2) one must rewrite the *problem* $\text{And} \circ (\text{Eq})^{\otimes k}$ as a different decision tree over different inputs.

2 Definitions: Communication Problems and Oracles

We will use some non-standard definitions that are more natural for constant-cost problems. These definitions come from e.g. [5, 13, 12, 14, 8].

It is convenient to define a *communication problem* as a set $\mathcal{P}$ of Boolean matrices, closed under row and column permutations. The more standard definition has one fixed function $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ for each input size n, with communication matrix $M_f \in \{0,1\}^{2^n \times 2^n}$, whereas we will think of a communication problem $\mathcal{P}$ as possibly containing many different communication matrices $M \in \{0,1\}^{N \times N}$ on each domain size N. (In the adjacency-in-trees problem, Example 1, there are many different trees on N vertices, which define many different communication matrices.)

For a fixed matrix $M \in \{0,1\}^{N \times N}$ and parameter $\delta < 1/2$, we write $\mathsf{R}_\delta(M)$ for the two-way, public-coin randomized communication complexity of M. For a communication problem $\mathcal{P}$, we write $\mathsf{R}_\delta(\mathcal{P})$ as the function

$$N \mapsto \max\left\{\mathsf{R}_\delta(M) \;:\; M \in \mathcal{P}, M \in \{0,1\}^{N \times N}\right\} .$$

Then the class BPP^0 is the collection of communication problems $\mathcal{P}$ which satisfy $\mathsf{R}_{1/4}(\mathcal{P}) = O(1)$.

To define communication with oracles, we require the notion of a *query set*:

▶ **Definition 8** (Query Set). *A* query set $\mathcal{Q}$ *is a set of matrices closed under (1) taking submatrices; (2) permuting rows and columns; and (3) copying rows and columns. For any set of matrices* $\mathcal{M}$, *we write* $\mathsf{QS}(\mathcal{M})$ *for the closure of* $\mathcal{M}$ *under these operations.*

Observe that if $\mathsf{R}_{1/4}(\mathcal{P}) = O(1)$ then $\mathsf{R}_{1/4}(\mathsf{QS}(\mathcal{P})) = O(1)$, since constant-cost protocols are preserved by row and column copying as well as taking submatrices.

▶ **Definition 9** (Communication with oracles). *Let $\mathcal{P}$ be any communication problem, i.e. set of Boolean matrices. For any $N \times N$ matrix M with values in a set Λ, write $\mathsf{D}^{\mathcal{P}}(M)$ for the minimum cost of a two-way deterministic protocol computing M as follows. The protocol is a binary tree T where each leaf node v is assigned a value $\ell(v) \in \Lambda$, and each inner node v is assigned a query matrix $Q \in \{0,1\}^{N \times N}$ where $Q \in \mathsf{QS}(\mathcal{P})$. On any pair of inputs $(i,j) \in [N] \times [N]$, the protocol proceeds as follows: the current pointer c is initiated as the root of T, and at every step, if $Q_c(i,j) = 1$ then the pointer c moves to its left child, and otherwise if $Q_c(i,j) = 0$ then the pointer c moves to the right. Once the pointer c reaches a leaf, the output of the protocol is the value $\ell(c)$ assigned to the leaf c. It is required that $\ell(c) = M(i,j)$. The cost of the protocol is the depth of T.*

This definition differs from the standard definition of oracle communication because we do not restrict the input size of the oracle. Specifically, each oracle query is represented by an $N \times N$ matrix $Q \in \mathsf{QS}(\mathcal{P})$, obtained by taking a submatrix of an *arbitrarily large* instance of $P \in \mathcal{P}$ and then copying rows and columns. This is the natural definition because this preserves constant-cost randomized protocols, whereas preserving non-constant cost functions usually requires restricting the size of the instance $P \in \mathcal{P}$.

▶ Remark 10. For constant-cost communication problems, i.e. problems $\mathcal{P} \in \mathsf{BPP}^0$, we will simply identify the problem $\mathcal{P}$ with its query set $\mathsf{QS}(\mathcal{P})$ since this does not change the communication complexity of $\mathcal{P}$. For example, $\mathsf{D}^{\mathrm{EQ}}(\cdot)$ is $\mathsf{D}^{\mathcal{Q}}(\cdot)$ where $\mathcal{Q}$ is taken to be the closure $\mathsf{QS}(\{I_{N,N}\})$ of the identity matrices.

3 Better Boosting of Equality Protocols

We prove the first part of Theorem 4, that Equation (1) can be improved for the Equality oracle. This theorem will also be applied in the later sections of the paper.

The proof uses the "noisy-search-tree" argument of [10]. This is a well-known idea that was previously applied in [21] to get an upper bound on the communication complexity of Greater-Than; see also the textbook exercise in [23]. We only require the observation that the argument works for arbitrary Equality queries, not just the binary search queries used in those papers. Also, we did not find any complete exposition of the proof of the Greater-Than upper-bound: the application of [10] in [21] is black-box and informal, and the models of computation in these two works do not match up, which causes some very minor gaps in the proof[1], so we make an effort to give a complete exposition here.

Informal protocol sketch. The idea of the protocol is that an Equality-oracle protocol is a binary tree T, where each node is a query to the oracle. On any given input, there is one "correct" path through T. The randomized protocol keeps track of a current node c in the tree T. In each round, the node c either moves down to one of its children, or, if it detects that a mistake has been made in an earlier round, it moves back up the tree. There are two main ideas:

[1] The gap is that the outputs of the Equality subroutine are *not* independent random variables. As far as we can tell, this very minor issue persists in the textbook exercise in [23] devoted to the Greater-Than problem.

1. At every node c, the protocol can "double-check" the answers in *all* ancestor nodes with only $O(1)$ communication overhead, which *implicitly* reduces the error of all previous queries. This uses a special property of the EQUALITY oracle, that a conjunction of equalities $(a_1 = b_1) \wedge (a_2 = b_2) \wedge \cdots \wedge (a_t = b_t)$ is equivalent to a single equality $(a_1, a_2, \ldots, a_t) = (b_1, b_2, \ldots, b_t)$. We can use this property to check if the current node c is on the "correct" path. (This simple observation is our contribution to this argument.)
2. The random walk of the node c through the tree is likely to stay close to the "correct" path; this is essentially the argument of [10].

▶ **Theorem 11.** *For any $M \in \Lambda^{N\times N}$ with values in an arbitrary set Λ,*

$$\mathsf{R}_\delta(M) = O\left(\mathsf{D}^{\mathrm{EQ}}(M) + \log\frac{1}{\delta}\right).$$

Proof. Let T be the tree of depth $d = \mathsf{D}^{\mathrm{EQ}}(M)$ as in Definition 9. For a node v in T let $a_v, b_v \colon [N] \to \mathbb{N}$ be the functions defining the oracle query at the node v with $Q_v(i,j) = \mathrm{EQ}(a_v(i), b_v(j))$. Let $R := 4 \cdot \max\{d, C\log(1/\delta)\}$ where C is a sufficiently large constant, and construct a tree T' by replacing each leaf node v of T with another tree L_v of depth $C\log(1/\delta)$ (where C is a sufficiently large constant), with each node v' of L_v being a copy of the parent node of v in T (i.e. the functions $a_{v'}, b_{v'} \colon [N] \to \mathbb{N}$ are identical to those of the parent of v). We then simulate the protocol defined by T using Algorithm 1.

Algorithm 1 Noisy-Tree Protocol.

Input: Row i, column j of communication matrix M.

Initialize pointer $c \leftarrow \mathrm{root}(T')$.
for $r \in [R]$ **do**
  Let $P = (p_1, p_2, \ldots, p_k)$ be the path in T' from $\mathrm{root}(T')$ to c.
  Let $(q_1, q_2, \ldots, q_t)$ be the subsequence of P where the protocol has taken the left branch.
  ▷ *(i.e. the nodes where the protocol previously detected "equality".)*
  Use the EQUALITY protocol with error probability $1/4$ to check

$$(a_{q_1}(i), a_{q_2}(i), \ldots, a_{q_t}(i)) = (b_{q_1}(j), b_{q_2}(j), \ldots, b_{q_t}(j))?$$

  ▷ *Re-check all previous "equality" answers simultaneously.*
  if inequality is detected on the sequence $q_1, \ldots, q_t$ **then**
  ▷ *A mistake was detected in an earlier round; go back up.*
    Update c to be the parent of c in T'.
  else
  ▷ *Check the current node and continue.*
    Use the EQUALITY protocol with error probability $1/4$ to check $a_c(i) = b_c(j)$?
    if Equality is detected **then** move c to its left child, otherwise move c to its right child.
if c belongs to a subtree L_v (replacing leaf v of T) **then**
  return $\ell(v)$. Otherwise **return** 0.

Since Algorithm 1 performs R rounds using in each round at most 2 instances of the randomized EQUALITY protocol with error $1/4$, the total amount of communication is at most $O(R) = O(\max\{d, \log(1/\delta)\})$ as desired. Let us now verify correctness.

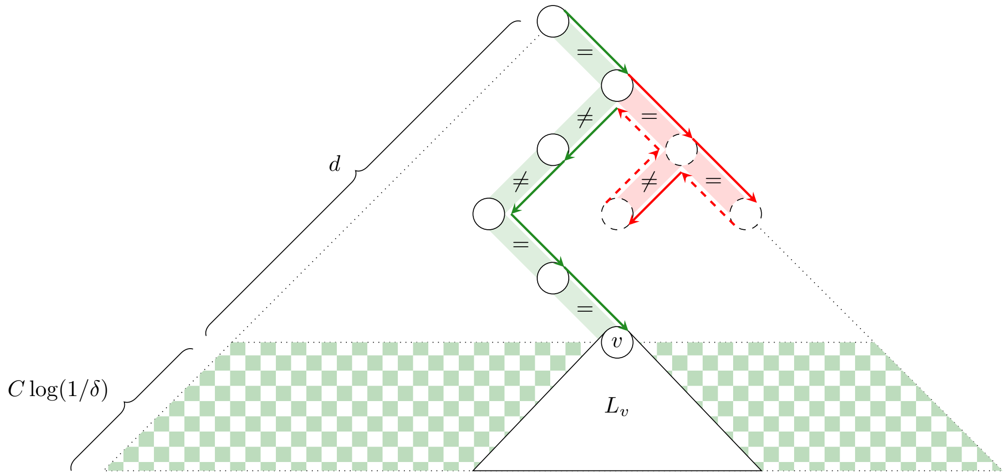

Figure 1 The picture represents the runtime of Algorithm 1. The thick green path is $P'_{i,j}$ for some i and j. The walk corresponding to the runtime of Algorithm 1 is represented with thin arrows: green arrows represent *good* rounds, solid red arrows represent bad rounds where protocol makes a mistake, and dashed red arrows represent bad rounds where protocol backtracks.

For any inputs i, j there is a unique root-to-leaf path $P_{i,j}$ taken in T ending at some leaf v, and a corresponding unique path $P'_{i,j}$ in T' which terminates at the subtree L_v. For any execution of Algorithm 1, we say a round r is "good" if the pointer c starts the round on a vertex in $P'_{i,j} \cup L_v$ and also ends the round on a vertex in $P'_{i,j} \cup L_v$. We say round r is "bad" otherwise. Write $\boldsymbol{g}$ for the number of good rounds and $\boldsymbol{b}$ for the number of bad rounds, which are random variables satisfying $R = \boldsymbol{b} + \boldsymbol{g}$.

▷ Claim 12. If $\boldsymbol{g} > d$ then the protocol produces a correct output.

Proof of claim. Observe that, if $c \in P'_{i,j}$ at the start of round r, then the counter cannot move back up, because the EQUALITY protocol has one-sided error and will correctly report that the concatenated strings are equal with probability 1. So the protocol must have terminated with the counter c at a descendent of the g^{th} node of $P'_{i,j}$. Since $\boldsymbol{g} > d$, the protocol terminated with c in the subtree of T' that replaced the final node v of $P_{i,j}$, meaning that it will output the correct value. ◁

We say that the protocol *makes a mistake* in round r if the randomized EQUALITY protocol erroneously outputs "equal" in Line 5 when $(a_{q_1}(i), \ldots, a_{q_t}(i)) \neq (b_{q_1}(j), \ldots, b_{q_t}(j))$, or if these tuples are truly equal but the protocol erroneously reports "equal" in Line 9 when $a_c(i) \neq b_c(j)$. Define the random variable $\boldsymbol{m}_r := 1$ if the protocol makes a mistake in round r and 0 otherwise, and define $\boldsymbol{m} = \sum_{r=1}^{R} \boldsymbol{m}_r$ for the total number of rounds where the protocol makes a mistake.

▷ Claim 13. $\boldsymbol{b} \leq 2\boldsymbol{m}$.

Proof of claim. Consider any bad round r. Either the counter c moves up or down the tree. If the counter c moves up to its parent c', then we charge the bad round to the most recent round $r' < r$ where the counter started at c' and observe that the protocol must have made a mistake at round r'. Otherwise, if the counter c moves down the tree, we charge the bad round to r itself and observe that the protocol makes a mistake in round r. Then we see that each round where a mistake is made is charged for at most 2 bad rounds (one for itself, if the counter moves down; and one for the earliest round where the counter returns to its current position). ◁

If the protocol outputs the incorrect value then we must have $\boldsymbol{g} = R - \boldsymbol{b} \le d$ and therefore $R - d \le \boldsymbol{b} \le 2\boldsymbol{m}$, so $\boldsymbol{m} \ge \frac{R-d}{2}$. It remains to bound the number of mistakes $\boldsymbol{m}$; we write $\boldsymbol{m} = \sum_{r=1}^{R} \boldsymbol{m}_r$ where $\boldsymbol{m}_r$ indicates whether the protocol makes a mistake in round r.

In any round r, conditional an all previous rounds, the probability that the protocol makes a mistake is at most 1/4: either there is an ancestor node in P where a mistake was made in an earlier round, in which case a mistake is made in round r only if it makes an error in Line 5; or the path P is entirely correct and the protocol makes a mistake only if there is an error in Line 9. So $\mathbb{P}[\boldsymbol{m}_r = 1 \mid \boldsymbol{m}_1, \ldots, \boldsymbol{m}_{r-1}] \le 1/4$ for every r and $\mu := \mathbb{E}[\boldsymbol{m}] \le R/4$. Using known concentration bounds (e.g. Theorem 3.1 of [17]), for any $\frac{1}{4} \le \gamma \le 1$ we have $\mathbb{P}[\boldsymbol{m} \ge \gamma R] \le e^{-R \cdot D(\gamma \| \delta)}$; in particular, since $R = 4 \cdot \max\{d, C\log(1/\delta)\}$, we have $\frac{R-d}{2} \ge \frac{3R}{8}$, so for constant $\kappa := D\left(\frac{3}{8} \| \frac{1}{4}\right) > 0$,

$$\mathbb{P}\left[\boldsymbol{m} \ge \frac{R-d}{2}\right] \le \mathbb{P}\left[\boldsymbol{m} \ge \frac{3}{8} \cdot R\right] \le e^{-R \cdot \kappa} \le e^{-4C\log(1/\delta) \cdot \kappa} \le \delta\,,$$

when we choose C to be a sufficiently large constant. ◀

4 No Better Boosting for Hamming Distance, and Consequences

We now complete the proof of Theorem 4 by showing that Equation (1) is nearly tight for the 1-Hamming Distance oracle. We prove this with a direct-sum result, showing that computing k independent copies of 1-Hamming Distance cannot be computed without the $\log k$-factor loss from boosting. Let us define the direct sum problems.

For any function $f \colon X \times Y \to Z$ and any $k \in \mathbb{N}$, we define function $f^{\otimes k}$ as the function which computes k copies of f, i.e. $f^{\otimes k} \colon X^k \times Y^k \to Z^k$ where on inputs $x \in X^k$ and $y \in Y^k$,

$$f^{\otimes k}(x, y) = (f(x_1, y_1), f(x_2, y_2), \ldots, f(x_k, y_k))\,.$$

It is easy to see that $\mathsf{D}^{\mathrm{HD}_1}((\mathrm{HD}_1^n)^{\otimes k}) = k$ for $n > 1$ since we can compute each copy of HD_1^n with one query. In this section we prove:

▶ **Theorem 14.** *For all $n \ge 4k^2$, $\mathsf{R}_{1/4}((\mathrm{HD}_1^n)^{\otimes k}) = \Omega(k \log k / \log\log k)$. Consequently, there exist matrices M such that*

$$\mathsf{R}_{1/4}(M) = \Omega\left(\frac{\mathsf{D}^{\mathrm{HD}_1}(M) \cdot \log \mathsf{D}^{\mathrm{HD}_1}(M)}{\log\log \mathsf{D}^{\mathrm{HD}_1}(M)}\right)\,.$$

Our proof has two further consequences. The first is about randomized parity decision trees (see e.g. [4] for definitions and background on parity decision trees): it is not hard to see that the randomized parity decision tree complexity of the 1-Hamming Weight function $\mathrm{HW}_1^n \colon \{0,1\}^n \to \{0,1\}$ defined by $\mathrm{HW}_1^n(x) = 1$ iff $|x| = 1$ is $\mathsf{RPDT}(\mathrm{HW}_1^n) = O(1)$. Since $\mathrm{HD}_1^n(x, y) = \mathrm{HW}_1^n(x \oplus y)$ and one can simulate each parity query with two bits of communication, we get $\mathsf{R}_{1/4}((\mathrm{HD}_1^n)^{\otimes k}) = O(\mathsf{RPDT}_{1/4}((\mathrm{HW}_1^n)^{\otimes k}))$. Together these statements imply:

▶ **Corollary 15.** *For $n \ge 4k^2$, $\mathsf{RPDT}((\mathrm{HW}_1^n)^{\otimes k}) = \Omega(k \log k / \log\log k)$.*

The second consequence of our proof, explained in Section 4.3, is the optimal $\Omega(\log n)$ lower bound on the number of Equality queries required to compute HD_1^n. All of these results come from our main lemma, a randomized reduction from HD_k^n to $(\mathrm{HD}_1^n)^{\otimes O(k)}$.

4.1 Randomized Reduction Lemma

▶ **Lemma 16.** *For $c = 9/10$, and for all $k \in \mathbb{N}$, let $R = \log_{1/c} k$ and $\delta := \frac{1}{10R}$. Then*

$$\mathsf{R}_{1/4}(\mathrm{HD}_k^n) = O\left(\sum_{i=0}^{R} \mathsf{R}_\delta((\mathrm{HD}_1^n)^{\otimes(4k\cdot c^i)})\right).$$

Proof. Our protocol for $\mathrm{HD}_k^n(x, y)$ is Algorithm 2. Let $c = 9/10$ and let C be some constant to be determined later. For a string $x \in \{0,1\}^n$ and a set $S \subseteq [n]$, we will write $x_S \in \{0,1\}^{|S|}$ for the substring of x on coordinates S.

Algorithm 2 Hamming Distance Reduction.

Input: $x, y \in \{0,1\}^n$.

1: Initialize $T \leftarrow [n]$; $\ell \leftarrow k$.
2: **while** $\ell > C$ **do**
3: Let $S_1, \dots, S_{4\ell}$ be a uniformly random partition of T.
4: Let $u_i = x_{S_i}$; $v_i = y_{S_i}$ be the substrings of x, y on subsets S_i for all $i \in [4\ell]$.
5: Run δ-error protocols for $(\mathrm{HD}_1^n)^{\otimes 4\ell}((u_1, v_1), \dots, (u_{4\ell}, v_{4\ell}))$ and $\mathrm{EQ}_n^{\otimes 4\ell}((u_1, v_1), \dots, (u_{4\ell}, v_{4\ell}))$.
▷ *Assuming these subroutines are correct, we know* $\mathsf{dist}(u_i, v_i)$ *exactly, if* $\mathsf{dist}(u_i, v_i) \in \{0, 1\}$.
6: $w_i \leftarrow \mathsf{dist}(u_i, v_i)$ if $\mathsf{dist}(u_i, v_i) \leq 1$ and 2 otherwise.
▷ *We can safely output 0 if we see more than ℓ differences:*
7: **if** $\sum_{i \in [4\ell]} w_i > \ell$ **then return** 0.
▷ *In the next step, isolate the sets S_i where the protocol finds exactly one difference.*
8: $s \leftarrow |\{i \in [4\ell] \mid w_i = 1\}|$.
▷ *If* $\mathsf{dist}(x_T, y_T) > \ell$*, we should see many sets with exactly one difference; output 1 otherwise:*
9: **if** $s < \ell/10$ **then return** 1.
▷ *Throw out sets S_i with at most one difference; update the number ℓ of remaining differences.*
10: $T \leftarrow \bigcup_{i \in [4\ell]\,:\, w_i = 2} S_i$.
11: $\ell \leftarrow \ell - s$.
12: **return** $\mathrm{HD}_\ell^{|T|}(x_T, y_T)$.

First, let us calculate the cost of the protocol. As guaranteed by Line 9, at each iteration the value of ℓ is reduced to at most $\frac{9}{10}\ell = c\ell$, so there are at most $R = \log_{1/c} k$ iterations, and in the i-th iteration (indexed from zero), $\ell \leq kc^i$. Hence, at each iteration, the communication cost is at most

$$\mathsf{R}_\delta((\mathrm{HD}_1^n)^{\otimes 4kc^i}) + \mathsf{R}_\delta((\mathrm{EQ})^{\otimes 4kc^i}) \leq 2 \cdot \mathsf{R}_\delta((\mathrm{HD}_1^n)^{\otimes 4kc^i}).$$

Since C is a constant, the cost of the final step with $\ell \leq C$ is $O(1)$.

Now let us estimate the error. Since there are at most $R = \log_{1/c} k$ iterations and the 2 protocols in Line 5 each have error at most $\delta = 1/10R$, the total probability of an error occurring in Line 5 is at most $1/5$. We may therefore assume from now on the perfect correctness of the values w_i.

Under this assumption, the protocol maintains the invariant that the number of bits outside T where x, y differ is $\mathsf{dist}(x_{[n]\setminus T}, y_{[n]\setminus T}) = k - \ell$, so it cannot output the incorrect value in Line 7. Let us consider the probability that the protocol outputs the incorrect value in Line 9. This only occurs if $\mathsf{dist}(x_T, y_T) > \ell$ and $s < \ell/10$. We need to estimate $\mathbb{P}[|\{i \in [4\ell] \mid w_i = 1\}| \geq \ell/10]$. The size of the set $\{i \in [4\ell] \mid w_i > 0\}$ is the number of unique colors we get when coloring each element i of the set $\Delta_T := \{i \in T : x_i \neq y_i\}$ of cardinality $|\Delta_T| = \mathsf{dist}(x_T, y_T)$ uniformly with color $\boldsymbol{\chi}_i \sim [4\ell]$; call this number $\boldsymbol{\chi} := |\{\boldsymbol{\chi}_i : i \in \Delta_T\}|$. We know that Line 9 does not halt, so $|\{i \in [4\ell] \mid w_i = 2\}| < \ell/2$. Then, if $\boldsymbol{\chi} \geq (6/10)\ell$, it must be that $|\{i \in [4\ell] : w_i = 1\}| \geq \boldsymbol{\chi} - \ell/2 \geq \ell/10$, so Line 9 does not halt. For simplicity, since $|\Delta_T| \geq \ell$, in the next expression we consider only the first ℓ elements of Δ_T and identify them with the set $[\ell]$. The probability we need to estimate is

$$\mathbb{P}[|\{\boldsymbol{\chi}_i \mid i \in [\ell]\}| \leq 0.6\ell] \leq \sum_{S \in \binom{[\ell]}{0.6\ell}} \mathbb{P}[\{\boldsymbol{\chi}_i \mid i \in [\ell]\} \subseteq \{\boldsymbol{\chi}_i \mid i \in S\}]$$
$$\leq \binom{\ell}{0.6\ell} \cdot \left(\frac{6}{10 \cdot 4}\right)^{0.4\ell} < 2^{\ell} \cdot 2^{\log_2(3/20) \cdot 0.4\ell} \leq 2^{-.01\ell}.$$

We have that the total error is bounded by $\sum_{\ell=C}^{\infty} 2^{-.01\ell} \leq 2^{-.01C}/(1 - 2^{-.01}) \leq 100 \cdot 2^{-.01C}$, so choosing C to be large enough we get arbitrarily small constant error. ◀

4.2 Direct Sum Theorem for 1-Hamming Distance

We require the lower bound on the communication cost of HD_k^n:

▶ **Theorem 17** ([24]). *For all* $k^2 \leq \delta n$, $\mathsf{R}_\delta(\mathrm{HD}_k^n) = \Omega(k \log(k/\delta))$.

Now we can prove Theorem 14.

Proof of Theorem 14. Assume for contradiction that $\mathsf{R}_{1/4}((\mathrm{HD}_1^n)^{\otimes k}) = o(k \log k / \log \log k)$, so by standard boosting,

$$\mathsf{R}_\delta((\mathrm{HD}_1^n)^{\otimes k}) = o\left(\frac{k \log k}{\log \log k} \cdot \log \frac{1}{\delta}\right).$$

Then by Lemma 16, with $c = 9/10$ and $R = \log_{1/c} k$,

$$\mathsf{R}_{1/4}(\mathrm{HD}_k^n) = O\left(\sum_{i=0}^{R} \mathsf{R}_\delta((\mathrm{HD}_1^n)^{4kc^i})\right) = \sum_{i=0}^{R} o\left(\frac{kc^i \log(kc^i)}{\log \log(kc^i)} \log \log k\right)$$
$$= \sum_{i=0}^{R} o(c^i k \log k) = o(k \log k),$$

which contradicts Theorem 17 when $n \geq 4k^2$. ◀

Our Corollary 15 for randomized parity decision trees follows easily from this theorem since a randomized parity decision tree for 1-Hamming Weight, (or k copies of it), can be simulated by a randomized communication protocol to compute 1-Hamming Distance (or k copies of it).

4.3 Lower Bound on Computing 1-Hamming Distance with Equality Queries

Recently, [12, 13] showed that Equality is not complete for the class BPP^0 of constant-cost communication problems, and [8] showed that there is *no* complete problem for this class. The independent and concurrent proofs of [12, 13] both showed that $\mathsf{D}^{\mathrm{EQ}}(\mathrm{HD}_1^n) = \omega(1)$.

We showed above that functions which reduce to EQUALITY have better boosting, while 1-HAMMING DISTANCE does not, so 1-HAMMING DISTANCE cannot reduce to EQUALITY – this gives a new and unexpected proof that EQUALITY is not complete for BPP^0:

▶ **Corollary 18.** $\mathsf{D}^{\mathrm{EQ}}(\mathrm{HD}_1^n) = \omega(1)$. *Therefore, EQUALITY is not a complete problem for* BPP^0.

There is an easy upper bound of $\mathsf{D}^{\mathrm{EQ}}(\mathrm{HD}_1^n) = O(\log n)$ obtained using binary search. With a more careful argument we can strengthen the above result and get a new proof that this is optimal, matching the lower bound already given in [12] by Fourier analysis.

▶ **Theorem 19.** $\mathsf{D}^{\mathrm{EQ}}(\mathrm{HD}_1^n) = \Theta(\log n)$.

Proof. Assume for the sake of contradiction that $\mathsf{D}^{\mathrm{EQ}}(\mathrm{HD}_1^n) = o(\log n)$, which immediately implies $\mathsf{D}^{\mathrm{EQ}}((\mathrm{HD}_1^n)^{\otimes k}) = o(k \log n)$. By Theorem 11 we then have $\mathsf{R}_\delta((\mathrm{HD}_1^n)^{\otimes k}) \le o(k \log n) + O(\log 1/\delta)$. Applying Lemma 16 we get, for $c = 9/10$ and $\delta = \frac{1}{10 \log_{1/c} k}$,

$$\mathsf{R}_{1/4}(\mathrm{HD}_k^n) = O\left(\sum_{i=0}^{\log_{1/c} k} \mathsf{R}_\delta((\mathrm{HD}_1^n)^{\otimes 4kc^i})\right)$$
$$= \sum_{i=0}^{\log_{1/c} k} (o(kc^i \log n) + O(\log\log k)) = o(k \log n) + O(\log k \log\log k).$$

Applying this inequality with $n = k^4$ we get $\mathsf{R}_{1/4}(\mathrm{HD}_k^{k^4}) = o(k \log k)$, which contradicts Theorem 17. ◀

▶ Remark 20. It is interesting that the additive $O(\log(1/\delta))$ in Theorem 11 is required for this proof. If the $\log(1/\delta)$ term was multiplicative, we would get a bound of $o(k \log n \cdot \log\log k)$ in the sum, giving $o(k \log k \log\log k)$ when we set $n = k^4$, which is not in contradiction with Theorem 17. So the weaker (but still non-trivial) bound $\mathsf{R}_{1/4}(M) = O(\mathsf{D}^{\mathrm{EQ}}(M))$ would not suffice, although it would still allow us to conclude $\mathsf{D}^{\mathrm{EQ}}(\mathrm{HD}_1^n) = \omega(1)$. The trivial bound of $\mathsf{R}_{1/4}(M) = O(\mathsf{D}^{\mathrm{EQ}}(M) \log \mathsf{D}^{\mathrm{EQ}}(M))$ would not allow us to prove even $\mathsf{D}^{\mathrm{EQ}}(\mathrm{HD}_1^n) = \omega(1)$.

5 Noisy-Tree Fails for Other Oracles

At this point we cannot determine whether better boosting is possible *only* for the constant-cost protocols which reduce to EQUALITY. But we can make some progress towards this question by observing that the "noisy-tree" protocol in Theorem 11 does not work for any other oracles in BPP^0. To state this formally, we must define a reasonable generalization of that protocol.

The noisy-tree protocol relied on two properties of the EQUALITY oracle. The first is that it has one-sided error (the protocol for EQUALITY will output the correct answer with probability 1 when the inputs are equal). The second property is what we will call the *conjunction property*:

A query set $\mathcal{Q}$ has the *conjunction property* if there exists a constant c such that for all $d \in \mathbb{N}$ and all $Q_1, \dots, Q_d \in \mathcal{Q}$, $\mathsf{D}^{\mathcal{Q}}\left(\bigwedge_{i=1}^d Q_i\right) \le c$ where $\bigwedge_{i=1}^d Q_i$ denotes the problem of computing

$$Q_1(x_1, y_1) \wedge Q_2(x_2, y_2) \wedge \cdots \wedge Q_d(x_d, y_d)\,.$$

on d pairs of inputs $(x_1, y_1), \dots, (x_d, y_d)$. For example, EQUALITY has the conjunction property because computing

$$\mathrm{EQ}(x_1, y_1) \wedge \mathrm{EQ}(x_2, y_2) \wedge \cdots \wedge \mathrm{EQ}(x_d, y_d)$$

can be done with the single query $\textsc{Eq}\left(((x_1, x_2, \ldots, x_d), (y_1, y_2, \ldots, y_d)\right)$. Following the proof of Theorem 11, we could claim the following result, which would hold even for oracles $\mathcal{Q}$ that have non-constant cost (but still using arbitrary-size oracle queries[2]):

▶ **"Theorem".** Let $\mathcal{Q}$ be any query set satisfying the conjunction property, and whose elements $Q \in \mathcal{Q}$ admit one-sided error randomized communication protocols with cost $O(\mathsf{R}(\mathcal{Q}))$. Then for any $M \in \{0,1\}^{N \times N}$, $\mathsf{R}_\delta(M) = O(\mathsf{D}^{\mathcal{Q}}(M) \cdot \mathsf{R}_{1/4}(\mathcal{Q}) + \log \frac{1}{\delta})$.

But it turns out that this does not really generalize Theorem 11, even if we require only the conjunction property (i.e. ignore one-sided error):

▶ **Proposition 21.** *If $\mathcal{Q}$ is a query set that satisfies the conjunction property, then it is either a subset of the query set of* Equality*, or it is the set of all matrices.*

To prove this, we use VC dimension. The VC dimension of a Boolean matrix M is the largest d such that there are d columns of M, where the submatrix of M restricted to those columns contains all 2^d possible distinct rows. A family $\mathcal{F}$ of matrices has *bounded VC dimension* if there is a constant d such that all $M \in \mathcal{F}$ have VC dimension at most d. If $\mathcal{F}$ is closed under taking submatrices (and permutations), then it has bounded VC dimension if and only if it is not the family of all matrices.

Proof of Proposition 21. If $\mathcal{Q}$ does not contain the matrix $\left[\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, then it is not hard to see that $\mathcal{Q}$ is a subset of the query set of Equality. So we suppose that $\mathcal{Q}$ contains the matrix $\left[\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right]$ and satisfies the conjunction property. We first show:

▷ Claim 22. Every matrix $M \in \{0,1\}^{N \times N}$ is a submatrix of $\bigwedge_{i=1}^N Q_i$ where each $Q_i = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right]$.

Proof of claim. Let each Q_i be a copy of $\left[\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, so that $Q = \bigwedge_{i=1}^N Q_i$ has row space $[2]^N$ and column space $[2]^N$. Let $M \in \{0,1\}^{N \times N}$ and map each row $x \in [N]$ of M to the row $v(x) \in [2]^N$ of Q with

$$\forall j \in [N] : v(x)_j = \begin{cases} 1 & \text{if } M(x,j) = 1 \\ 2 & \text{if } M(x,j) = 0\,, \end{cases}$$

and map each column $y \in [N]$ of M to the column $w(y) \in [2]^N$ of Q with

$$\forall j \in [N] : w(y)_j = \begin{cases} 1 & \text{if } j \neq y \\ 2 & \text{if } j = y\,. \end{cases}$$

For any row x and column y of M, if $M(x,y) = 1$ then

$$Q_i(v(x)_j, w(y)_j) = \begin{cases} Q_i(1,1) = 1 & \text{if } M(x,j) = 1 \text{ and } j \neq y \\ Q_i(1,2) = 1 & \text{if } M(x,j) = 1 \text{ and } j = y \\ Q_i(2,1) = 1 & \text{if } M(x,j) = 0 \text{ and } j \neq y\,. \end{cases}$$

This covers all the cases, since we never have $M(x,j) = 0$ and $j = y$, so $Q(v(x), w(y)) = 1$. Finally, if $M(x,y) = 0$ then

$$Q_i(v(x)_y, w(y)_y) = Q_i(2,2) = 0\,,$$

so $Q(v(x), w(y)) = 0$. Therefore M is a submatrix of Q. ◁

---

[2] Arbitrary-size oracle queries may be sensible for non-constant cost problems that still have bounded VC dimension, e.g. Greater-Than oracles as in [5].

By the conjunction property, there is a constant c such that $\mathsf{D}^{\mathcal{Q}}(M) \leq \mathsf{D}^{\mathcal{Q}}(\bigwedge_{i=1}^{N} Q_i) \leq c$ for all $M \in \{0,1\}^{N \times N}$. Therefore, there is a constant C and a function $f \colon \{0,1\}^C \to \{0,1\}$ such that all matrices M can be written as

$$M(x,y) := f(Q_1(x,y), Q_2(x,y), \dots, Q_C(x,y))$$

where each $Q_i \in \mathcal{Q}$ (think of f as the function which simulates the protocol for $\mathsf{D}^{\mathcal{Q}}(M)$ using the answers to each query Q_i; see e.g. [14] for the simple proof of this fact). Let $f(\mathcal{Q})$ denote the set of all matrices which can be achieved in this way, which we have argued is the set of all matrices. For the sake of contradiction, assume that $\mathcal{Q}$ is not the set of all matrices, so that the VC dimension $\mathsf{VC}(\mathcal{Q})$ is bounded. Then standard VC dimension arguments (see e.g. [20]) show that the VC dimension of $f(\mathcal{Q})$ is at most $O(\mathsf{VC}(\mathcal{Q}) \cdot C \log C)$. Since C is constant, the VC dimension of $f(\mathcal{Q})$ is therefore also bounded, but $f(\mathcal{Q})$ contains all matrices, so this is a contradiction and $\mathcal{Q}$ must contain all matrices. ◀

▶ Remark 23. If one is interested only in constant-cost oracles, we may replace the conjunction property $\mathsf{D}^{\mathcal{Q}}(\bigwedge_i Q_i) \leq c$ with the property $\mathsf{R}_{1/4}\left(\bigwedge_{i=1}^{d} Q_i\right) = O(1)$, but the same proof rules out this generalization as well.

References

1 Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. *SIAM Journal on Computing*, 42(3):1327–1363, 2013. doi:10.1137/100811969.

2 Shalev Ben-David, Mika Göös, Robin Kothari, and Thomas Watson. When is amplification necessary for composition in randomized query complexity? In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2020.

3 Eric Blais and Joshua Brody. Optimal separation and strong direct sum for randomized query complexity. In *34th Computational Complexity Conference (CCC 2019)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2019.

4 Arkadev Chattopadhyay, Ankit Garg, and Suhail Sherif. Towards stronger counterexamples to the log-approximate-rank conjecture. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, 2021.

5 Arkadev Chattopadhyay, Shachar Lovett, and Marc Vinyals. Equality alone does not simulate randomness. In *34th Computational Complexity Conference (CCC 2019)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019.

6 Louis Esperet, Nathaniel Harms, and Andrey Kupavskii. Sketching distances in monotone graph classes. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

7 Yuting Fang, Mika Göös, Nathaniel Harms, and Pooya Hatami. Constant-cost communication does not reduce to k-hamming distance, 2024. arXiv:2407.20204.

8 Yuting Fang, Lianna Hambardzumyan, Nathaniel Harms, and Pooya Hatami. No complete problem for constant-cost randomized communication. In *Proceedings of the Symposium on Theory of Computing (STOC 2024)*, 2024.

9 Tomás Feder, Eyal Kushilevitz, Moni Naor, and Noam Nisan. Amortized communication complexity. *SIAM Journal on computing*, 24(4):736–750, 1995. doi:10.1137/S0097539792235864.

10 Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994. doi:10.1137/S0097539791195877.

11 Lianna Hambardzumyan, Hamed Hatami, and Pooya Hatami. A counter-example to the probabilistic universal graph conjecture via randomized communication complexity. *Discrete Applied Mathematics*, 322:117–122, 2022. doi:10.1016/J.DAM.2022.07.023.

12 Lianna Hambardzumyan, Hamed Hatami, and Pooya Hatami. Dimension-free bounds and structural results in communication complexity. *Israel Journal of Mathematics*, 253(2):555–616, 2022.

13 Nathaniel Harms, Sebastian Wild, and Viktor Zamaraev. Randomized communication and implicit graph representations. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2022)*, pages 1220–1233, 2022. doi:10.1145/3519935.3519978.

14 Nathaniel Harms and Viktor Zamaraev. Randomized communication and implicit representations for matrices and graphs of small sign-rank. In *Proceedings of the Symposium on Discrete Algorithms (SODA)*. SIAM, 2024.

15 Hamed Hatami and Pooya Hatami. Guest column: Structure in communication complexity and constant-cost complexity classes. *ACM SIGACT News*, 55(1):67–93, 2024. doi:10.1145/3654780.3654788.

16 Hamed Hatami, Pooya Hatami, William Pires, Ran Tao, and Rosie Zhao. Lower bound methods for sign-rank and their limitations. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

17 Russell Impagliazzo and Valentine Kabanets. Constructive proofs of concentration bounds. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 617–631. Springer, 2010. doi:10.1007/978-3-642-15369-3_46.

18 Mauricio Karchmer, Ran Raz, and Avi Wigderson. Super-logarithmic depth lower bounds via the direct sum in communication complexity. *Computational Complexity*, 5:191–204, 1995. doi:10.1007/BF01206317.

19 Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1996.

20 Jiri Matousek. *Lectures on discrete geometry*, volume 212. Springer Science & Business Media, 2013.

21 Noam Nisan. The communication complexity of threshold gates. *Combinatorics, Paul Erdos is Eighty*, 1:301–315, 1993.

22 Ramamohan Paturi and Janos Simon. Probabilistic communication complexity. *Journal of Computer and System Sciences*, 33(1):106–123, 1986. doi:10.1016/0022-0000(86)90046-2.

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

24 Mert Sağlam. Near log-convexity of measured heat in (discrete) time and consequences. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 967–978. IEEE, 2018.