A Candidate for a Strong Separation of Information and Communication

Mark Braverman*[1], Anat Ganor†[2], Gillat Kol[3], and Ran Raz‡[4]

1 Department of Computer Science, Princeton University, USA
2 Department of Computer Science, Tel-Aviv University, Israel
3 Department of Computer Science, Princeton University, USA
4 Department of Computer Science, Princeton University, USA

Abstract

The *weak interactive compression conjecture* asserts that any two-party communication protocol with communication complexity C and information complexity I can be compressed to a protocol with communication complexity $poly(I)polylog(C)$.

We describe a communication problem that is a candidate for refuting that conjecture. Specifically, while we show that the problem can be solved by a protocol with communication complexity C and information complexity $I = polylog(C)$, the problem seems to be hard for protocols with communication complexity $poly(I)polylog(C) = polylog(C)$.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases communication complexity, amortized communication complexity, communication compression, direct sum, information complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.11

1 Introduction

The classical data compression theorem shows that every message can be compressed to its information content, measured using the entropy function. Can one prove a similar result in the interactive setting, where two parties engage in an interactive communication protocol? That is, can the transcript of every communication protocol be compressed to (roughly) its "information content" [2]?

The information content of an interactive protocol is typically measured using the *information complexity* measure [11, 16, 7, 1, 2]. In this paper we will mainly be interested in *internal* information complexity (a.k.a, information complexity and information cost). A related notion of *external* information complexity is also used in the literature. Roughly speaking, let π be a two-party communication protocol, and let μ be a distribution over the private inputs for the communicating parties. The (internal) information complexity of π over μ, denoted $\mathsf{IC}_\mu(\pi)$, is the number of information bits that the players learn about each other's input, when running the protocol π with inputs distributed according to μ (see Definition 3).

* Research supported in part by NSF Awards DMS-1128155, CCF- 1525342, and CCF-1149888, a Packard Fellowship in Science and Engineering, and the Simons Collaboration on Algorithms and Geometry.
† Research supported by the Israel Science Foundation (grant number 552/16) and the I-CORE Program of the planning and budgeting committee and The Israel Science Foundation (grant number 4/11).
‡ Research supported by the Simons Collaboration on Algorithms and Geometry and by the National Science Foundation grants No. CCF-1714779 and CCF-1412958.

9th Innovations in Theoretical Computer Science Conference (ITCS 2018).
Editor: Anna R. Karlin; Article No. 11; pp. 11:1–11:13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Using the notion of information complexity, the above interactive compression problem can be formulated as asking whether for every protocol π and distribution μ with information complexity $I = IC_\mu(\pi)$, there exists a "compressed" protocol π' that produces (almost) the same output as π and has $CC_\mu^{avg}(\pi')$ close to I. Here, $CC_\mu^{avg}(\pi')$ stands for the *distributional communication complexity* of π' over μ, which is the expected number of bits communicated by π' when the inputs to the players are sampled according to μ (see Definition 2).

Several recent results show how to compress communication protocols in several cases, starting from [2] (see Section 2.3). However, none of these results gives a way of compressing a general protocol to a protocol that only communicates I bits, or even $poly(I)$ bits. We note that in some special cases, compression to $poly(I)polylog(C)$ or even $poly(I)$ are known to be possible (see Section 2.3).

The difficulty in compressing general protocols was recently explained by the authors, by proving exponential gaps between the distributional communication complexity and information complexity of some carefully designed communication tasks. In [8, 10], Ganor, Kol and Raz showed an explicit example of a boolean function with (internal) information complexity $\leq I$ and distributional communication complexity $\geq 2^{\Omega(I)}$ (see [17] for a simplified proof). In [9], Ganor, Kol and Raz analyzed a communication task proposed by Braverman [4], with (external) information complexity $\leq I$ and distributional communication complexity $\geq 2^{\Omega(I)}$.

One drawback of these results is that the protocols that achieve information complexity I have communication complexity double or even triple exponential in I. Therefore, while these results rule out "strong" compression to $poly(I)$, they leave open the possibility of "weak" compression to $poly(I)polylog(C)$.

► **Open Problem 1.** *Is it true that for every computational task f, distribution μ over the inputs and every communication protocol π that solves f with error $o(1)$, there exists a protocol π' that solves f with error $o(1)$, such that*

$$CC_\mu^{avg}(\pi') \leq poly(IC_\mu(\pi)) \cdot polylog(CC_\mu^{avg}(\pi))?$$

A general compression to $poly(I)polylog(C)$ as suggested by Problem 1, if exists, still yields very efficient compressed protocols that potentially constitute huge savings. Due to the equivalence between interactive compression and direct sum [2, 5], such a compression would also imply a near optimal *direct sum* result for distributional communication complexity, thus resolve this long standing open problem in the affirmative. Specifically, it will show that the distributional communication complexity of solving m independent copies of a communication task is almost as high as m times the distributional communication complexity of solving a single copy. Moreover, such an interactive compression result gives rise to a new paradigm for protocol design, where one is only mindful to the information revealed by the protocol, and then uses a compression scheme as a "black-box" to lower the required communication complexity.

In this work we suggest a candidate communication problem, called the *excited tree game*, for ruling out the $poly(I)polylog(C)$ compression scheme suggested by Problem 1. The game is defined in Section 3, and is parameterized by a parameter $c \in \mathbb{N}$. In Section 5, we construct a protocol for solving the game with information complexity $I = polylog(c)$ and communication complexity $C = O(c)$. In Section 4, we try to justify the conjecture that there is no protocol for solving the game with distributional communication complexity at most $poly(I)polylog(C) = polylog(c)$. Observe that this conjecture, if true, shows that the low information protocol we construct for the excited tree game cannot be compressed to $poly(I)polylog(C)$, thus answers Problem 1 in the negative. Proving this conjecture in full, however, seems very challenging.

2 Preliminaries

2.1 Communication Complexity

In the two player distributional model of communication complexity, each player gets an input, where the inputs are sampled from a joint distribution that is known to both players. The players' goal is to solve a computational task that depends on both inputs. The players can use both common and private random strings and are allowed to err with some small probability. The players communicate in rounds, where in each round one of the players sends a message to the other player. The communication complexity of a protocol is the total number of bits communicated by the two players. The communication complexity of a computational task is the minimum number of bits that the players need to communicate in order to solve the task with high probability, where the minimum is taken over all protocols. For excellent surveys on communication complexity see [14, 15]. In this work it would be more convenient to work with average communication complexity.

▶ **Definition 2 (Average Communication Complexity).** The *average communication complexity* of a protocol π over random inputs (X, Y) that are drawn according to a joint distribution μ, denoted $CC_\mu^{avg}(\pi)$, is the expected number of communication bits transmitted during the protocol, where the expectation is over (X, Y) and over the randomness. The ϵ average communication complexity of a computational task f with respect to a distribution μ is defined as

$$CC_\mu^{avg}(f, \epsilon) = \inf_\pi CC_\mu^{avg}(\pi),$$

where the infimum ranges over all protocols π that solve f with error at most ϵ on inputs that are sampled according to μ.

2.2 Information Complexity

Roughly speaking, the (internal) information complexity of a protocol is the number of information bits that the players learn about each other's input, when running the protocol. The information complexity of a communication task is the minimum number of information bits that the players learn about each other's input when solving the task, where the minimum is taken over all protocols. Formally,

▶ **Definition 3 (Information Cost).** The *information cost* of a protocol π over random inputs (X, Y) that are drawn according to a joint distribution μ, is defined as

$$IC_\mu(\pi) = \mathbf{I}(\Pi; X|Y) + \mathbf{I}(\Pi; Y|X),$$

where Π is a random variable which is the transcript of the protocol π with respect to μ. That is, Π is the concatenation of all the messages exchanged during the execution of π. The ϵ information cost of a computational task f with respect to a distribution μ is defined as

$$IC_\mu(f, \epsilon) = \inf_\pi IC_\mu(\pi),$$

where the infimum ranges over all protocols π that solve f with error at most ϵ on inputs that are sampled according to μ.

2.3 Known Compression Protocols

Several beautiful recent results show how to compress communication protocols in several cases. Barak, Braverman, Chen and Rao showed how to compress any protocol with information complexity I and communication complexity C, to a protocol with communication complexity $\sqrt{I \cdot C} \cdot polylog(C)$ [2]. They also suggest a protocol that communicates $I^{ext} \cdot polylog(C)$ bits, where I^{ext} is the external information complexity of the original protocol. Braverman and Rao showed how to compress any one round (or constant number of rounds) protocol with information complexity I to a protocol with communication complexity $O(I)$ [5]. Braverman showed how to compress any protocol with information complexity I to a protocol with communication complexity $2^{O(I)}$ [3] (see also [6, 12]). Building over [2], Kol and Sherstov showed how to compress any protocol with information complexity I to a protocol with communication complexity $I \cdot polylog(I)$ in the case where the underlying distribution is a product distribution [13, 18].

3 The Excited Tree Game

The excited tree game is a communication game for two players A and B. The game is played on a rooted, complete, binary tree $\mathcal{T}$, of depth c, where c is larger than a sufficiently large constant. Player A "owns" every non-leaf vertex in even layers and player B "owns" every non-leaf vertex in odd layers. For every non-leaf vertex v, the owner of v gets as an input a distribution $P_v = (p_v, 1 - p_v)$ and the other player gets as an input a distribution $Q_v = (q_v, 1 - q_v)$, both distributions are over the children of v. We think of every P_v as the "correct" distribution over the two children of v. The distributions $\{P_v, Q_v\}_v$ are chosen in a very specific way that is described below.

A *frontier* in the tree is a set of vertices that contains exactly one vertex (leaf or non-leaf) on every path from the root to a leaf. Given a vertex v and a frontier S in the tree, we say that v is *above* the frontier S if on the path from the root to v there is no vertex in S. We say that v is *on* the frontier if v is in S. If v is neither above the frontier nor on it, then it is *below* the frontier.

We denote by x, y the inputs to the players A, B respectively. That is, x is the set of all the distributions P_v or Q_v that are given to player A and y is the set of all the distributions P_v or Q_v that are given to player B. We define the distribution μ on the inputs to the players by an algorithm for sampling an input pair (x, y) (Algorithm 1 below).

Fix some $k = polylog(c)$ such that $\log^4(c) \le k$. Let μ_1 be the uniform distribution over the interval $[-\frac{k}{\sqrt{c}}, \frac{k}{\sqrt{c}}]$ and let μ_2 be the uniform distribution over the interval $[-\frac{1}{\sqrt[8]{c}}, \frac{1}{\sqrt[8]{c}}]$.[1] In Algorithm 1 below, we sample for every non-leaf vertex v two values $x_1(v), x_2(v)$ according to μ_1, μ_2 respectively. Next, when we say "set v to be non-excited", we mean "set $p_v = \frac{1}{2} + x_1(v) + x_2(v)$ and $q_v = \frac{1}{2} + x_2(v) - x_1(v)$". By "set v to be excited", we mean "set $p_v = \frac{1}{2} + x_1(v) + x_2(v)$ and $q_v = \frac{1}{2} + x_1(v) - x_2(v)$". Note that without communication, none of the players can distinguish between an excited vertex and a non-excited vertex, since p_v and q_v have the same distribution in both cases.

Given the distributions P_v for every non-leaf vertex v and the frontier S in the tree, we define a distribution P_S over the vertices in S. For every vertex $w \in S$, let $v_0, v_1, ..., v_\ell$ be

[1] One can also consider other symmetric distributions in the range $[-1, 1]$ with expectation 0 and variance $O(\frac{k^2}{c}), O(\frac{1}{\sqrt[4]{c}})$ respectively, as well as other values for k, and changing the interval $[-\frac{1}{\sqrt[8]{c}}, \frac{1}{\sqrt[8]{c}}]$ to $[-\frac{1}{c^\beta}, \frac{1}{c^\beta}]$, for some other $0 < \beta < 1/2$.

Algorithm 1 Sample (x, y) according to μ

1. For every non-leaf vertex v we sample two values $x_1(v), x_2(v)$ according to μ_1, μ_2 respectively.
2. Let S be a frontier in the tree defined as follows: Pick every vertex to be in S, independently, with probability $\alpha = \frac{k}{c}$. Then, for every path from the root to a leaf, remove from S all vertices on that path, except for the vertex closest to the root, if such a vertex exists. If there is no vertex in S on a path from the root to a leaf, add that leaf to S.
3. Set every non-leaf vertex above the frontier S to be non-excited.
4. Set every non-leaf vertex on the frontier S or below it to be excited.

the vertices on the path in the tree from the root to w, where ℓ is the layer of w. That is, v_0 is the root, $v_\ell = w$ and for every $0 \le i < \ell$, v_{i+1} is a child of v_i. Then, $P_S(w)$ is obtained by sampling every child on the path to w according to the correct distribution of its parent. That is,

$$P_S(w) = \prod_{i=0}^{\ell-1} P_{v_i}(v_{i+1}) \text{ where } P_{v_i}(v_{i+1}) = \begin{cases} p_{v_i} & \text{if } v_{i+1} \text{ is the left hand child of } v_i \\ 1 - p_{v_i} & \text{if } v_{i+1} \text{ is the right hand child of } v_i \end{cases}.$$

The players' mutual goal is to output the same vertex w on the frontier S, where S is the frontier defined in Algorithm 1, such that for almost all possible outputs w, the probability that they both output w is close to $P_S(w)$. More precisely, let x, y be the inputs to the players A, B respectively and let μ be the distribution over the inputs. Let $A(x, y), B(x, y)$ denote the output values of A, B respectively. Note that $A(x, y), B(x, y)$ are random variables that depend on the randomness. For a communication protocol π, we say that π solves the game with respect to μ with error ϵ if

$$\Pr[A(x, y) = B(x, y)] \ge 1 - \epsilon \quad \text{and} \quad \mathbb{E}\left[||A(x, y) - P_S||_1\right] \le \epsilon,$$

where the probability is over inputs that are sampled according to μ and over the randomness, the expectation is over inputs that are sampled according to μ and $|| \cdot ||_1$ is the ℓ_1 norm. Note that $A(x, y)$ is referred to as a distribution as well as a random variable and that the distribution P_S depends on x and y.

In Section 5 we prove the following lemma.

▶ **Lemma 4.** *There exists a protocol that solves the excited tree game with respect to μ with error $o(1)$, with average communication complexity $O(c)$ and information complexity polylog(c).*

Therefore, to answer Open Problem 1 in the negative, it is enough to answer the following question affirmatively.

▶ **Open Problem 5.** *Is it true that for $\epsilon = o(1)$ the ϵ average communication complexity of the excited tree game with respect to μ is at least $(\log(c))^{\omega(1)}$?*

4 Why Excited Tree?

At a very high level, the excited tree game can be viewed as follows. The game is played on a rooted, complete, binary tree $\mathcal{T}$, of depth c. A frontier S is chosen in $\mathcal{T}$. All the vertices above the frontier are set to be "non-excited" and the vertices below the frontier are set to be "excited". The player's goal is to output a vertex on the frontier S, sampled according to the "correct" distribution.

Let us start with the rational behind the name *excited tree game*. In physics, an "excited" state is a state with a higher energy level than the ground ("non-excited") state. In the excited tree game, an excited vertex is a vertex with a higher "information level" than a non-excited vertex. For an excited vertex v the distance between the distributions P_v and Q_v is large and hence the information that the player who doesn't own v is missing is relatively large. For a non-excited vertex v the distance between the distributions P_v and Q_v is small and hence the information that the player who doesn't own v is missing is relatively small.

Since all the vertices above the frontier are non-excited, there is a relatively simple protocol with low information complexity for the excited tree game: Starting from the root, until reaching the frontier, at every vertex v, the player owning v samples a child of v according to P_v and sends a bit b_v to the other player, to indicate which child was sampled. Both players continue to the child of v that is indicated by the communicated bit. Since all the vertices above the frontier are non-excited, the information given by each bit b_v is small and hence the entire information complexity of the protocol is small. The only complication in this protocol is that the players have to stop when they reach the frontier. We show how to do that while keeping the information complexity of the protocol low.

To answer Open Problem 5 affirmatively, one needs to prove a lower bound of $(\log(c))^{\omega(1)}$ on the communication complexity of the excited tree game. While we don't have such a proof, we note that several approaches to solve the game with communication complexity $(\log(c))^{O(1)}$, seem to fail.

Two properties of the excited tree game that makes it difficult (or impossible...) to solve with low communication complexity are as follows:

1. Without communication, none of the players can distinguish between an excited vertex and a non-excited vertex, since p_v and q_v have the same distribution in both cases. Hence, without communication (or with relatively small communication) the players don't have a lot of information about which vertices are above the frontier and which vertices are below the frontier.
2. For every vertex v above the frontier, the restriction of the inputs of the two players to the subtree below v (conditioned on the event that v is above the frontier) has the same distribution as the distribution of the excited tree game played on a smaller tree. In fact, we could have defined the problem on an *infinite*, rooted, complete binary tree, and then the distribution of the restriction to the subtree below v (conditioned on the event that v is above the frontier) would have been exactly the same as the original distribution. (We chose to work with a finite tree for simplicity of the presentation).

In light of these properties, let us consider a few approaches for designing protocols with low communication complexity for solving the problem, based on known approaches for compression protocol.

A first approach (inspired by ideas initiated in [2] and used in many subsequent works) could be to try to simulate the above mentioned low-information protocol, by starting from the root and trying to sample, according to the correct distribution, vertices that are lower and lower in the tree, until reaching the frontier. A major difficulty with such attempts is the second property above. By the second property, even if the two players managed to agree on a vertex v above the frontier, sampled according to the correct distribution, they still have to solve a copy of pretty much the same problem as the one that they started with, and hence they made no (or very little) progress. The two players only make progress if the vertex v that they agreed on happens to be exactly on the frontier.

A second approach (inspired by ideas initiated in [2, 5, 3] and used in many subsequent works) could be to sample a leaf in the tree (or a vertex which is with high probability below the frontier) and climb up from that vertex to the frontier. A major difficulty with such attempts is that all the vertices below the frontier are excited, and hence the distributions that the two players have on the leaves are very far from each other, so it's hard for them to agree on a leaf. They could agree on a leaf sampled according to a pre-agreed distribution, known to both players, such as the uniform distribution, and climb up from that leaf to the frontier. However, that would not sample a frontier vertex according to the correct distribution. In general, the first property above (that the players don't know where the frontier is) makes such attempts very difficult.

A third approach that one may consider for attacking this and related problems (and that, to the best of our knowledge, has not been used before), is to try to sample a vertex v above the frontier (as in the first approach) and from that vertex to move down to the closest frontier vertex u in the subtree below v. This approach is based on the fact that there should be a frontier vertex at a distance of roughly $\log(\alpha^{-1}) = O(\log(c))$ below v. A difficulty with such attempts is that it is not clear how to find the closest frontier vertex u by a protocol with small communication complexity.

We note that turning these intuitions and ideas into a full proof for a lower bound on the communication complexity of the excited tree game seems very challenging.

5 Information Upper Bound

In this section, we prove Lemma 4. Let $(x, y) \in \operatorname{supp}(\mu)$ be an input pair for the excited tree game and let S be the frontier defined in Algorithm 1. Let π be the following protocol for the excited tree game, played on the input pair (x, y): Starting from the root, at every vertex v, the player owning v samples a child of v according to P_v and sends a bit b_v to the other player, to indicate which child was sampled. Both players continue to the child of v that is indicated by the communicated bit.

After receiving a bit b_v, the receiving party, without loss of generality the second player, sends a bit a_v, that supposedly indicates whether the players are above the frontier S or not, where $a_v = 1$ stands for "below or on the frontier" and $a_v = 0$ stands for "above the frontier". If v is a leaf, the second player sends $a_v = 1$. Otherwise, to determine the value of a_v, the second player considers the last $\ell = 4k\sqrt[4]{c}$ vertices $v_1, \dots, v_\ell$ reached by the protocol and owned by the first player and the corresponding bits $b_{v_1}, \dots, b_{v_\ell}$ that were sent by the first player (if less than ℓ bits were sent by the first player so far, the second player sends $a_v = 0$). For every $j \in [\ell]$, the second player compares b_{v_j} and $\tilde{q_{v_j}}$, where $\tilde{q_{v_j}}$ is 1 if $q_{v_j} \geq \frac{1}{2}$ and 0 otherwise. The second player sends $a_v = 1$ if less than $\frac{\ell}{2}$ of these pairs are equal, and otherwise, he sends $a_v = 0$.

Once the bit $a_v = 1$ was sent, the players run a binary search over the last 3ℓ vertices reached by the protocol, with the goal of finding the vertex on the frontier (if less than 3ℓ vertices were reached by the protocol so far, the binary search is over all the vertices reached by the protocol). In each iteration of the binary search, the players send their input distributions corresponding to the current vertex considered by the binary search. The probabilities are truncated so that each player sends k bits per vertex. For each such vertex v, the players calculate $|p'_v - q'_v|$, where p'_v, q'_v are the truncated p_v, q_v respectively. The binary search assumes that $|p'_u - q'_u| \leq \frac{3k}{\sqrt{c}}$ for all the vertices u among these 3ℓ vertices that are above the frontier, and that $|p'_u - q'_u| > \frac{3k}{\sqrt{c}}$ for all the vertices u among these 3ℓ vertices that are below the frontier. Under this assumption, the players output the vertex v which is the first vertex among these 3ℓ vertices for which $|p'_v - q'_v| > \frac{3k}{\sqrt{c}}$, if such a vertex exists. (Otherwise, the players output an error message).

5.1 Bounding the Error Probability

In Claims 6 and 7 we prove that with high probability, the bit $a_v = 1$ is sent below or on the frontier, but not too far below it. In Claim 8 we prove that if this is the case, then with high probability, the players output the vertex on the frontier reached by the protocol. Note that each vertex reached by the protocol is chosen according to the correct distribution. That is, if a vertex w was reached by the protocol, then when the players reached its parent v, they sampled w according to the distribution P_v. Therefore, Claims 6, 7 and 8 imply that the distribution of the vertex output by the players is close to the goal distribution P_S.

▶ **Claim 6.** *Let $(x, y) \in \operatorname{supp}(\mu)$ be an input pair for the excited tree game and let S be the frontier defined in Algorithm 1. Then, with probability at least $1 - c \cdot 2^{-k/2}$ over the input pair (x, y) and over the randomness, the bit $a_v = 1$ is sent when the players are below or on the frontier S.*

Proof. Let v be a non-excited vertex. First, consider the case that $p_v \geq \frac{1}{2}$. In this case,

$$\begin{aligned}\Pr\left[q_v < \tfrac{1}{2} \mid p_v \geq \tfrac{1}{2}\right] &= \Pr\left[x_2(v) - x_1(v) < 0 \mid x_1(v) + x_2(v) \geq 0\right] \\ &= \Pr\left[x_2(v) < x_1(v) \mid x_2(v) \geq -x_1(v)\right] \\ &= \Pr\left[(x_2(v) < x_1(v)) \wedge (x_2(v) \geq -x_1(v))\right] \cdot \left(\Pr\left[x_2(v) \geq -x_1(v)\right]\right)^{-1} \\ &= 2\Pr\left[-x_1(v) \leq x_2(v) < x_1(v)\right] \\ &\leq 2\Pr\left[-\frac{k}{\sqrt{c}} \leq x_2(v) \leq \frac{k}{\sqrt{c}}\right] = \frac{2k}{c^{3/8}}.\end{aligned}$$

Therefore, with high probability, $q_v \geq \frac{1}{2}$ and $\tilde{q_v} = 1$. It holds that

$$\begin{aligned}\Pr\left[b_v = \tilde{q_v} \mid \left(p_v \geq \tfrac{1}{2}\right) \wedge \left(q_v \geq \tfrac{1}{2}\right)\right] &= \Pr\left[b_v = 1 \mid \left(p_v \geq \tfrac{1}{2}\right) \wedge \left(q_v \geq \tfrac{1}{2}\right)\right] \\ &= \mathbf{E}\left[p_v \mid \left(p_v \geq \tfrac{1}{2}\right) \wedge \left(q_v \geq \tfrac{1}{2}\right)\right],\end{aligned}$$

where the last equality holds since the probability is over the inputs and over the randomness. Bounding the expectation we get that

$$\begin{aligned}\mathbf{E}\left[p_v \mid \left(p_v \geq \tfrac{1}{2}\right) \wedge \left(q_v \geq \tfrac{1}{2}\right)\right] &= \mathbf{E}\left[p_v \mid (x_2(v) \geq -x_1(v)) \wedge (x_2(v) \geq x_1(v))\right] \\ &= \mathbf{E}\left[p_v \mid x_2(v) \geq |x_1(v)|\right] \\ &= \frac{1}{2} + \mathbf{E}\left[x_1(v) \mid x_2(v) \geq |x_1(v)|\right] + \mathbf{E}\left[x_2(v) \mid x_2(v) \geq |x_1(v)|\right] \\ &\geq \frac{1}{2} - \frac{k}{\sqrt{c}} + \mathbf{E}\left[x_2(v) \mid x_2(v) \geq |x_1(v)|\right] \\ &\geq \frac{1}{2} - \frac{k}{\sqrt{c}} + \mathbf{E}\left[x_2(v) \mid x_2(v) \geq 0\right] = \frac{1}{2} + \frac{1}{2c^{1/8}} - \frac{k}{\sqrt{c}}.\end{aligned}$$

Similarly, when $p_v < \frac{1}{2}$, the probability that $q_v \geq \frac{1}{2}$ is at most $\frac{2k}{c^{3/8}}$. Therefore, with high probability $q_v < \frac{1}{2}$ and

$$\begin{aligned}\Pr\left[b_v = \tilde{q_v} \mid \left(p_v < \tfrac{1}{2}\right) \wedge \left(q_v < \tfrac{1}{2}\right)\right] &= \mathbf{E}\left[1 - p_v \mid \left(p_v < \tfrac{1}{2}\right) \wedge \left(q_v < \tfrac{1}{2}\right)\right] \\ &\geq \frac{1}{2} - \frac{k}{\sqrt{c}} - \mathbf{E}\left[x_2(v) \mid x_2(v) < -|x_1(v)|\right] \\ &\geq \frac{1}{2} + \frac{1}{2c^{1/8}} - \frac{k}{\sqrt{c}}.\end{aligned}$$

Put together, we get that for a non-excited vertex v, the probability that $b_v = \tilde{q_v}$ is at least

$$\left(1 - \frac{2k}{c^{3/8}}\right) \cdot \left(\frac{1}{2} + \frac{1}{2c^{1/8}} - \frac{k}{\sqrt{c}}\right) \geq \frac{1}{2} + \frac{1}{4c^{1/8}}.$$

When the players are above the frontier, all the vertices reached by the protocol are non-excited. If a player considers ℓ non-excited vertices $v_1, \ldots, v_\ell$ and their corresponding bits $b_{v_1}, \ldots, b_{v_\ell}$, then by Chernoff, the probability that less than $\frac{\ell}{2}$ of the pairs $b_{v_j}, \tilde{q_{v_j}}$ are equal is at most $e^{-2\ell/16c^{1/4}} \le 2^{-k/2}$. That is, for every vertex v above the frontier, the probability that the bit $a_v = 1$ is sent is at most $2^{-k/2}$. Thus, by the union bound, the total probability that a bit $a_v = 1$ is sent above the frontier is at most $c \cdot 2^{-k/2}$. ◀

▶ **Claim 7.** *Let $(x, y) \in \mathrm{supp}(\mu)$ be an input pair for the excited tree game and let S be the frontier defined in Algorithm 1. Assume that the players are below or on the frontier. Then, with probability at least $1 - 2^{-k/2}$ over the input pair (x, y) and over the randomness, a player will send the bit $a_v = 1$ after at most 2ℓ steps.*

Proof. Let v be an excited vertex. First, consider the case that $p_v \ge \frac{1}{2}$. In this case,

$$\begin{aligned}
\Pr\left[q_v \ge \tfrac{1}{2} \mid p_v \ge \tfrac{1}{2}\right] &= \Pr\left[x_1(v) - x_2(v) \ge 0 \mid x_1(v) + x_2(v) \ge 0\right] \\
&= \Pr\left[x_1(v) \ge x_2(v) \mid x_2(v) \ge -x_1(v)\right] \\
&= \Pr\left[(x_1(v) \ge x_2(v)) \wedge (x_2(v) \ge -x_1(v))\right] \cdot \left(\Pr\left[x_2(v) \ge -x_1(v)\right]\right)^{-1} \\
&= 2\Pr\left[-x_1(v) \le x_2(v) \le x_1(v)\right] \\
&\le 2\Pr\left[-\frac{k}{\sqrt{c}} \le x_2(v) \le \frac{k}{\sqrt{c}}\right] = \frac{2k}{c^{3/8}}.
\end{aligned}$$

Therefore, with high probability, $q_v < \frac{1}{2}$ and $\tilde{q_v} = 0$. It holds that

$$\begin{aligned}
\Pr\left[b_v \ne \tilde{q_v} \mid \left(p_v \ge \tfrac{1}{2}\right) \wedge \left(q_v < \tfrac{1}{2}\right)\right] &= \Pr\left[b_v = 1 \mid \left(p_v \ge \tfrac{1}{2}\right) \wedge \left(q_v < \tfrac{1}{2}\right)\right] \\
&= \mathbf{E}\left[p_v \mid \left(p_v \ge \tfrac{1}{2}\right) \wedge \left(q_v < \tfrac{1}{2}\right)\right],
\end{aligned}$$

where the last equality holds since the probability is over the inputs and over the randomness. Bounding the expectation we get that

$$\begin{aligned}
\mathbf{E}\left[p_v \mid \left(p_v \ge \tfrac{1}{2}\right) \wedge \left(q_v < \tfrac{1}{2}\right)\right] &= \mathbf{E}\left[p_v \mid (x_2(v) \ge -x_1(v)) \wedge (x_2(v) > x_1(v))\right] \\
&\ge \frac{1}{2} - \frac{k}{\sqrt{c}} + \mathbf{E}\left[x_2(v) \mid x_2(v) \ge |x_1(v)|\right] \\
&\ge \frac{1}{2} - \frac{k}{\sqrt{c}} + \mathbf{E}\left[x_2(v) \mid x_2(v) \ge 0\right] = \frac{1}{2} + \frac{1}{2c^{1/8}} - \frac{k}{\sqrt{c}}.
\end{aligned}$$

Similarly, when $p_v < \frac{1}{2}$, the probability that $q_v < \frac{1}{2}$ is at most $\frac{2k}{c^{3/8}}$. Therefore, with high probability $q_v \ge \frac{1}{2}$ and

$$\begin{aligned}
\Pr\left[b_v \ne \tilde{q_v} \mid \left(p_v < \tfrac{1}{2}\right) \wedge \left(q_v \ge \tfrac{1}{2}\right)\right] &= \mathbf{E}\left[1 - p_v \mid \left(p_v < \tfrac{1}{2}\right) \wedge \left(q_v \ge \tfrac{1}{2}\right)\right] \\
&\ge \frac{1}{2} - \frac{k}{\sqrt{c}} - \mathbf{E}\left[x_2(v) \mid x_2(v) \le -|x_1(v)|\right] \\
&\ge \frac{1}{2} + \frac{1}{2c^{1/8}} - \frac{k}{\sqrt{c}}.
\end{aligned}$$

Put together, we get that for an excited vertex v, the probability that $b_v \ne \tilde{q_v}$ is at least

$$\left(1 - \frac{2k}{c^{3/8}}\right) \cdot \left(\frac{1}{2} + \frac{1}{2c^{1/8}} - \frac{k}{\sqrt{c}}\right) \ge \frac{1}{2} + \frac{1}{4c^{1/8}}.$$

If the players take 2ℓ steps after they reached an excited vertex, then the player who should send either $a_v = 0$ or $a_v = 1$ considers ℓ excited vertices $v_1, \ldots, v_\ell$ and their corresponding bits $b_{v_1}, \ldots, b_{v_\ell}$. By Chernoff, the probability that less than $\frac{\ell}{2}$ of the pairs $b_{v_j}, \tilde{q_{v_j}}$ are not equal is at most $e^{-2\ell/16c^{1/4}} \le 2^{-k/2}$. That is, the probability that the player sends $a_v = 0$ is at most $2^{-k/2}$. ◀

▶ **Claim 8.** *Let $(x, y) \in \mathrm{supp}(\mu)$ be an input pair for the excited tree game and let S be the frontier defined in Algorithm 1. Assume that the bit $a_v = 1$ is sent when the players are below or on the frontier S and not more than 2ℓ steps after the players reached a vertex on S. Let w be the vertex on S that they reached. Then, with probability at least $1 - \frac{48k^2}{c^{1/8}}$ over the input pair (x, y) and over the randomness, the players output the vertex w.*

Proof. Let v be a non-excited vertex. Recall that p'_v, q'_v are the truncated probabilities p_v, q_v respectively. It holds that $|p'_v - q'_v| \leq 2|x_1(v)| + 2^{-k}$, which is at most $\frac{3k}{\sqrt{c}}$ (with probability 1). For an excited vertex v, it holds that $|p'_v - q'_v| \geq 2|x_2(v)| - 2^{-k}$. The probability that $2|x_2(v)| - 2^{-k}$ is at most $\frac{3k}{\sqrt{c}}$ is less than $\frac{4k}{c^{3/8}}$. Taking a union bound over the 3ℓ vertices considered by the binary search, we get that with probability of at least $1 - \frac{48k^2}{c^{1/8}}$, for all the excited vertices u among the 3ℓ vertices considered by the binary search, we have that $|p'_u - q'_u| > \frac{3k}{\sqrt{c}}$.

Thus, with probability of at least $1 - \frac{48k^2}{c^{1/8}}$, we have that $|p'_u - q'_u| \leq \frac{3k}{\sqrt{c}}$ for all the vertices u among the 3ℓ vertices considered by the binary search, that are above the frontier, and $|p'_u - q'_u| > \frac{3k}{\sqrt{c}}$ for all the vertices u among these 3ℓ vertices that are below the frontier. Under this assumption, the binary search outputs the vertex v which is the first vertex among these 3ℓ vertices for which $|p'_v - q'_v| > \frac{3k}{\sqrt{c}}$. Therefore, with probability at least $1 - \frac{48k^2}{c^{1/8}}$, the players output the vertex w. ◀

5.2 Bounding the Information Cost

To upper bound the information cost of the protocol π we will use the method described in [8], that is based on the notion of divergence cost of a tree [2, 5].

▶ **Definition 9 (Relative Entropy).** Let $\phi_1, \phi_2 : \Omega \to [0, 1]$ be two distributions, where Ω is finite. The *relative entropy* between ϕ_1 and ϕ_2, denoted $\mathbf{D}(\phi_1 \| \phi_2)$, is defined as

$$\mathbf{D}(\phi_1 \| \phi_2) = \sum_{x \in \Omega} \phi_1(x) \log\left(\frac{\phi_1(x)}{\phi_2(x)}\right).$$

▶ **Definition 10 (Divergence Cost [2, 5]).** Consider a binary tree $\mathcal{T}$ whose root is r and distributions $P_v = (p_v, 1 - p_v), Q_v = (q_v, 1 - q_v)$ for every non-leaf vertex v in the tree. We think of P_v and Q_v as distributions over the two children of the vertex v. We define the *divergence cost* of the tree $\mathcal{T}$ recursively, as follows. $\mathbf{D}(\mathcal{T}) = 0$ if the tree has depth 0, otherwise,

$$\mathbf{D}(\mathcal{T}) = \mathbf{D}(P_r \| Q_r) + \mathop{\mathbf{E}}_{v \sim P_r}[\mathbf{D}(\mathcal{T}_v)], \tag{1}$$

where for every vertex v, $\mathcal{T}_v$ is the subtree of $\mathcal{T}$ whose root is v.

An equivalent definition of the divergence cost of $\mathcal{T}$ is obtained by following the recursion in Equation (1) and is given by the following equation:

$$\mathbf{D}(\mathcal{T}) = \sum_{v \in V} \tilde{p}_v \cdot \mathbf{D}(P_v \| Q_v), \tag{2}$$

where V is the vertex set of $\mathcal{T}$ and for a vertex $v \in V$, $\tilde{p}_v$ is the probability to reach v by following the distributions P_v, starting from the root. Formally, if v is the root of the tree $\mathcal{T}$, then $\tilde{p}_v = 1$, otherwise,

$$\tilde{p}_v = \begin{cases} \tilde{p}_u \cdot p_u & \text{if } v \text{ is the left-hand child of } u \\ \tilde{p}_u \cdot (1 - p_u) & \text{if } v \text{ is the right-hand child of } u. \end{cases}$$

We will bound the information cost of the protocol until the bit $a_v = 1$ is sent, that is, until the players decide that they are below or on the frontier. Denote the protocol that starts as π but ends when the bit $a_v = 1$ is sent by π'. Note that after the bit $a_v = 1$ is sent in π, the players exchange at most $O(k \cdot \log(\ell))$ bits, which adds at most $O(k \cdot \log(\ell))$ bits of information.

We denote by $\mathcal{T}_{\pi'}$ the binary tree associated with π'. That is, every vertex v of $\mathcal{T}_{\pi'}$ corresponds to a possible transcript of π' and the two edges going out of v are labeled by 0 and 1, corresponding to the next bit to be transmitted. The vertices of the tree $\mathcal{T}_{\pi'}$ have the following structure: Every vertex v of $\mathcal{T}_{\pi'}$ corresponds to a vertex $\tilde{v}$ of $\mathcal{T}$, the binary tree on which the excited tree game is played. For a vertex v in an odd layer of $\mathcal{T}_{\pi'}$, the next bit to be transmitted by π' on the vertex v is $b_{\tilde{v}}$. For a vertex v in an even layer of $\mathcal{T}_{\pi'}$, the next bit to be transmitted by π' on the vertex v is $a_{\tilde{v}}$.

Every input pair $(x, y) \in \text{supp}(\mu)$ for the excited tree game, induces a distribution $P_v = (p_v, 1 - p_v)$ for every vertex v of the tree $\mathcal{T}_{\pi'}$, where p_v is the probability that the next bit transmitted by the protocol π' on the vertex v and inputs x, y is 0. Namely, if v is in an odd layer of $\mathcal{T}_{\pi'}$, the distribution P_v is the input distribution $P_{\tilde{v}}$ of the player that owns $\tilde{v}$. If v is in an even layer of $\mathcal{T}_{\pi'}$ then $P_v = (1, 0)$ when the player sending $a_{\tilde{v}}$ decides that the players are above the frontier and $P_v = (0, 1)$ when $a_{\tilde{v}} = 1$ is sent (note that given x, y and v this decision is deterministic).

For every vertex v of $\mathcal{T}_{\pi'}$, we define an additional distribution $Q_v = (q_v, 1-q_v)$ (depending on the input pair (x, y)). For a vertex v in an odd layer of $\mathcal{T}_{\pi'}$, the distribution Q_v is the input distribution $Q_{\tilde{v}}$ of the player that doesn't own $\tilde{v}$. If v is in an even layer of $\mathcal{T}_{\pi'}$ then $Q_v = (1 - \frac{1}{c}, \frac{1}{c})$.

For the rest of the section, we think of $\mathcal{T}_{\pi'}$ as the tree $\mathcal{T}_{\pi'}$ together with the distributions P_v and Q_v, for every vertex v in the tree $\mathcal{T}_{\pi'}$. In [8], Ganor, Kol and Raz showed that $IC_\mu(\pi') \leq \mathbf{E}[\mathbf{D}(\mathcal{T}_{\pi'})]$, where $\mathbf{D}(\mathcal{T}_{\pi'})$ is the divergence cost of the tree and the expectation is over the sampling of the inputs according to μ and over the randomness. Together with the following claim, we get that $IC_\mu(\pi') \leq O(k^2)$.

▶ **Claim 11.** *Let π' be the protocol that starts as π but ends when the bit $a_v = 1$ is sent. Let $\mathcal{T}_{\pi'}$ be the binary tree associated with π', together with the distributions P_v and Q_v for every vertex v in the tree $\mathcal{T}_{\pi'}$, as defined above. Then,*

$$\mathbf{E}[\mathbf{D}(\mathcal{T}_{\pi'})] = O(k^2),$$

where the expectation is over the inputs and over the randomness.

Proof. We bound the divergence cost separately for vertices in odd layers and for vertices in even layers. First, we sum over vertices in even layers. For every vertex v in an even layer of $\mathcal{T}_{\pi'}$, if $P_v = (1, 0)$ then $\mathbf{D}(P_v \| Q_v) = \log\left(\frac{1}{1-\frac{1}{c}}\right) = \log\left(1 + \frac{1}{c-1}\right) < \frac{2}{c}$. Since there are at most c such vertices on every path and the probability of reaching each vertex is at most 1, the sum in Equation (2) taken over vertices in even layers with $P_v = (1, 0)$ is at most $c \cdot \frac{2}{c} = 2$. If $P_v = (0, 1)$ then $\mathbf{D}(P_v \| Q_v) = \log\left(\frac{1}{\frac{1}{c}}\right) = \log(c) \leq O(k)$. Along each path, there is only one vertex v for which $P_v = (0, 1)$, the last vertex reached by the protocol π'.

Next, we sum over vertices in odd layers along an average path. Recall that each such vertex v corresponds to a vertex $\tilde{v}$ in $\mathcal{T}$. Let v be a vertex in an odd layer of $\mathcal{T}_{\pi'}$. It holds that $|p_v - q_v| \le \frac{1}{4}$ and $p_v \ge \frac{5}{16}$, and therefore, $\left|\frac{p_v - q_v}{p_v}\right| \le \frac{4}{5}$. By Taylor's expansion,

$$\begin{aligned}
-p_v \ln \frac{q_v}{p_v} &= -p_v \ln\left(1 - \frac{p_v - q_v}{p_v}\right) \\
&\le (p_v - q_v) + \sum_{i=2}^{\infty} \frac{|p_v - q_v|^i}{i \cdot p_v^{i-1}} \\
&\le (p_v - q_v) + \sum_{i=2}^{\infty} \frac{|p_v - q_v|^i}{2p_v^{i-1}} \\
&= (p_v - q_v) + \frac{(p_v - q_v)^2}{2\left(p_v - |p_v - q_v|\right)} \\
&\le (p_v - q_v) + O\left((p_v - q_v)^2\right).
\end{aligned}$$

Similarly, $-(1-p_v)\ln\frac{1-q_v}{1-p_v} \le (q_v - p_v) + O\left((p_v - q_v)^2\right)$. We get that

$$\mathbf{D}(P_v||Q_v) = -p_v \log \frac{q_v}{p_v} - (1-p_v)\log\frac{1-q_v}{1-p_v} \le O\left((p_v - q_v)^2\right).$$

For each non-excited vertex v it holds that $|p_v - q_v| \le \frac{2k}{\sqrt{c}}$ and therefore, the non-excited vertices add at most $O(k^2)$ to the divergence cost along any path. For each excited vertex v it holds that $|p_v - q_v| \le \frac{2}{\sqrt[8]{c}}$. By Claim 7, the probability that there are more than 3ℓ excited vertices on a path is at most $2^{-k/2} \le 1/c$. Therefore, along an average path, the expected number of excited vertices is at most $O(\ell) = O(k\sqrt[4]{c})$ and they add at most

$$O\left(\ell \cdot \left(\frac{2}{\sqrt[8]{c}}\right)^2\right) = O(k)$$

to the expected divergence cost. Put together, the expected divergence cost of π' is $O(k^2)$. ◀

References

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

2 Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. In *STOC*, pages 67–76, 2010.

3 Mark Braverman. Interactive information complexity. In *STOC*, pages 505–524, 2012.

4 Mark Braverman. A hard-to-compress interactive task? *In 51th Annual Allerton Conference on Communication, Control, and Computing*, 2013.

5 Mark Braverman and Anup Rao. Information equals amortized communication. In *FOCS*, pages 748–757, 2011.

6 Mark Braverman and Omri Weinstein. A discrepancy lower bound for information complexity. In *APPROX-RANDOM*, pages 459–470, 2012.

7 Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew Chi-Chih Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *FOCS*, pages 270–278, 2001.

8 Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of information and communication. In *FOCS*, 2014.

9 Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of communication and external information. In *STOC*, pages 977–986, 2016.

10 Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of information and communication for boolean functions. *Journal of the ACM*, 63(5):46:1–46:31, 2016.

11 Amiram H. Kaspi. Two-way source coding with a fidelity criterion. *IEEE Transactions on Information Theory*, 31(6):735–740, 1985.

12 Iordanis Kerenidis, Sophie Laplante, Virginie Lerays, Jérémie Roland, and David Xiao. Lower bounds on information complexity via zero-communication protocols and applications. In *FOCS*, pages 500–509, 2012.

13 Gillat Kol. Interactive compression for product distributions. In *STOC*, pages 987–998, 2016.

14 Eyal Kushilevitz and Noam Nisan. Communication complexity. *Cambridge University Press*, 1997.

15 Troy Lee and Adi Shraibman. Lower bounds in communication complexity. *Foundations and Trends in Theoretical Computer Science*, 3(4):263–398, 2009.

16 Alon Orlitsky and James R. Roche. Coding for computing. *IEEE Transactions on Information Theory*, 47(3):903–917, 2001 (Preliminary version at the IEEE International Symposium on Information Theory (ISIT) 1995, FOCS 1995).

17 Anup Rao and Makrand Sinha. Simplified separation of information and communication. *Electronic Colloquium on Computational Complexity (ECCC)*, 2015.

18 Alexander A. Sherstov. Compressing interactive communication under product distributions. In *FOCS*, pages 535–544, 2016.